République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université Yahia Fares de Médéa

Faculté des Sciences

Département Mathématiques et Informatique

Mémoire de Master

Filière : Informatique

Spécialité : Ingénierie des Systèmes et Technologies du Web

Thème

Utilisation de grands modèles de langage (LLMs)

pour la modélisation des utilisateurs et l'identification du contexte de recherche dans la recherche d'informations personnalisées.

Encadré par

Promoteur :M. Rafa Tahar
Co-Promoteur :M. Rahim Messaoud

Réalisé par

Safaa Fellague
Meriem Guesmia

Année universitaire : 2025–2026

Remerciements

Avant toute chose, nous rendons grâce à Allah Tout-Puissant pour Ses innombrables bienfaits et pour nous avoir accordé la santé, la patience et la volonté nécessaires à la réalisation de ce travail.

Ce mémoire est le fruit de plusieurs années d'efforts et de persévérance, au cours desquelles nous avons bénéficié du soutien et de l'aide de plusieurs personnes que nous tenons à remercier sincèrement.

Nous exprimons notre profonde gratitude à notre encadrant, ***M. Rafa Tahar****, pour la qualité de son encadrement, sa patience, sa rigueur scientifique et ses précieux conseils tout au long de l'élaboration de ce travail.*

Nous adressons également nos sincères remerciements à ***M. Rahim Messaoud*** *pour son soutien, sa disponibilité et son aide précieuse.*

Nous tenons à remercier chaleureusement ***M. ABDOUN Mohamed Nabil*** *pour avoir accepté de présider le jury de soutenance de ce mémoire.*

Nos remerciements s'adressent également à ***Mme SADOUKI Fatiha*** *pour avoir accepté d'examiner et d'évaluer ce travail.*

Enfin, nous exprimons notre gratitude à toutes les personnes qui ont contribué, de près ou de loin, à l'aboutissement de ce mémoire.

Dédicaces

Alhamdulilah, au commencement comme à la fin, Lui par la grâce duquel les choses se réalisent, par la bienveillance duquel s'ouvrent les portes du savoir et de la réussite. Nous Lui exprimons toute notre gratitude pour les bienfaits, l'assistance, la réussite et la facilité qu'Il nous a accordés.

À celle dont Allah a placé le Paradis sous les pieds, à la source de l'amour et à la chaleur de la tendresse, à ma très chère mère ***Khadidja****, je dédie ce travail avec toute la gratitude et la reconnaissance que porte mon cœur. Je prie Allah de la préserver et de lui accorder une longue vie dans la santé et le bonheur.*

À celui qui m'a appris la patience, la persévérance et l'autonomie, à ce soutien inébranlable qu'est mon père bien-aimé ***Rachid****, je dédie cette réussite en témoignage de mon profond respect, de mon affection et de ma reconnaissance. Puisse Allah le récompenser de la meilleure des récompenses.*

À mes frères ***Adem ,Majd*** *et ma sœurs* ***Lina****, compagnons de route et refuge du cœur, qui ont toujours été une source de soutien, de force et d'encouragement, je dédie ce travail avec toute ma fierté et ma gratitude.*

À mes amies fidèles ***Aicha, Meriem, Hadil, Safa*** *, qui ont embelli mes journées par leur soutien, leurs paroles bienveillantes et leur fidélité, et qui ont été présentes dans les moments de difficulté avant ceux de joie. je dédie ce travaille à vous tous,avec toute ma reconnaissance, mon amour et ma gratitude. réussite avec une affection sincère et éternelle.*

Meriem Guesmia

Dédicaces

Au nom d'Allah, le Tout Miséricordieux, le Très Miséricordieux.

Je rends grâce à Allah, le Tout-Puissant, pour m'avoir accordé la force, la patience et la persévérance nécessaires afin d'achever ce travail. C'est par Sa volonté que ce parcours a été possible et que cette réalisation a pu voir le jour.

Je dédie ce mémoire à mes chers parents, pour l'amour inconditionnel qu'ils m'ont toujours témoigné, pour leurs encouragements constants ainsi que pour le soutien précieux qu'ils m'ont apporté tout au long de mes études.

*À mon père, **Khaled**, qui a été un modèle de courage, de responsabilité et de sacrifice, et qui n'a jamais cessé de croire en moi, je dédie ce travail avec tout mon respect et ma profonde gratitude.*

*À ma mère, **Aïcha**, source inépuisable d'amour, de tendresse et de bienveillance, dont les prières et le soutien ont toujours illuminé mon chemin, je dédie ce mémoire avec toute mon affection et ma reconnaissance.*

Qu'ils trouvent dans ce travail le fruit de leurs efforts, de leur dévouement et de leurs sacrifices, ainsi que l'expression de mon profond respect, de ma gratitude infinie et de mon immense amour.

*À mes frères et sœurs, **Marwa, Khadija, Abdelghani, Hossam et Saif El Islam**, qui ont toujours été présents à mes côtés, je dédie ce travail en témoignage de mon amour, de mon attachement et de ma reconnaissance pour leur soutien constant.*

À mes amies, qui ont partagé avec moi les moments de joie comme les moments de difficulté, et qui ont toujours été une source d'encouragement, de confiance et de motivation, je dédie également ce travail avec une sincère affection et une profonde gratitude.

À ma chère binôme, pour sa coopération, sa patience, son soutien et les moments de partage tout au long de cette expérience. Je lui adresse mes sincères remerciements et toute ma gratitude. Enfin, je dédie ce mémoire à toutes les personnes qui ont contribué, de près ou de loin, à ma

réussite et à l'accomplissement de ce parcours académique.

Safaa Fellague

Résumé

La croissance des ressources numériques et la diversité des informations disponibles sur le Web rendent l'accès à des contenus pertinents de plus en plus complexe. Les systèmes classiques de recherche d'information reposent principalement sur l'appariement lexical entre les requêtes et les documents, sans prendre suffisamment en compte les préférences et le contexte de l'utilisateur. Cette limitation a favorisé l'émergence de la Recherche d'Information Personnalisée (RIP), qui vise à adapter les résultats aux besoins spécifiques de chaque utilisateur.

Ce mémoire propose une approche de recherche d'information personnalisée basée sur les Grands Modèles de Langage (LLMs) pour la modélisation des utilisateurs et l'identification du contexte de recherche. Le système proposé, nommé RIP-LLM, repose sur trois phases principales :

— **Phase 1 : Modélisation du profil utilisateur .** Un profil utilisateur est construit en combinant une dimension thématique à long terme, issue d'une extraction profonde par LLM, et une dimension comportementale à court terme capturant les interactions et l'historique récents de l'utilisateur.

— **Phase 2 : Identification du contexte de recherche.** Analyse de la requête et de l'historique récent afin de déterminer l'intention réelle de l'utilisateur et de réduire l'ambiguïté des requêtes.

— **Phase 3 : Reformulation et enrichissement contextuel des requêtes.** Les LLMs reformulent la requête en tenant compte du profil utilisateur et du contexte identifié afin de réduire l'ambiguïté et d'améliorer sa pertinence.

L'approche proposée met en évidence l'apport des Grands Modèles de Langage dans le domaine de la recherche d'information personnalisée. En exploitant le profil utilisateur et le contexte de recherche, le système RIP-LLM permet de générer des reformulations de requêtes plus adaptées aux besoins de l'utilisateur et d'améliorer la qualité du processus de recherche.

Mots-clés : Recherche d'Information Personnalisée (RIP), Grands Modèles de Langage (LLMs), Modélisation de l'Utilisateur, Contexte de Recherche, Profil Utilisateur , Reformulation de Requêtes.

Abstract

The growth of digital resources and the diversity of information available on the Web make access to relevant content increasingly challenging. Traditional information retrieval systems mainly rely on lexical matching between queries and documents, often ignoring user preferences and search context. This limitation has led to the emergence of Personalized Information Retrieval (PIR), which aims to adapt search results to individual user needs.

This thesis proposes a personalized information retrieval approach based on Large Language Models (LLMs) for user modeling and search context identification. The proposed system, called RIP-LLM, is organized into three main phases:

— **Phase 1: user profile modeling.** The user profile is built by integrating a long-term thematic component, extracted through in-depth analysis using a Large Language Model (LLM), and a short-term behavioral component that reflects the user's recent interactions, searches, and browsing history.

— **Phase 2: Search context identification.** Analyzing the query and recent search history to determine the user's actual intent and reduce ambiguity.

— **Phase 3: Context-aware query reformulation.** LLMs reformulate and enrich the query based on the user profile and the identified search context to improve its relevance.

The proposed approach highlights the contribution of Large Language Models (LLMs) in the field of Personalized Information Retrieval. By leveraging the user profile and the search context, the RIP-LLM system generates query reformulations that are better aligned with the user's information needs and improves the overall quality of the retrieval process.

Keywords: Personalized Information Retrieval (PIR), Large Language Models (LLMs), User Modeling, Search Context, User Profile, Query Reformulation.

Table des matières

LISTE DES FIGURES **VII**

LISTE DES TABLEAUX **IX**

LISTE DES SIGLES ET ACRONYMES **X**

INTRODUCTION GÉNÉRALE **XII**

Chapitre I : De la recherche d'information Classique à la personnalisation **1**

1 Introduction 2

2 Recherche d'Information Classique 2

2.1 Notions de base de la recherche d'information 3

2.1.1 Terme, document, requête et collection 3

2.1.2 Besoin d'information et pertinence 4

2.1.3 Système de Recherche d'Information (SRI) 5

2.2 Processus de recherche d'information 5

2.2.1 Expression du besoin en information 5

2.2.2 Processus d'indexation 5

2.2.3 Reformulation de la requête 8

2.2.4 Appariement document–requête 8

2.3 Modèles de recherche d'information 8

2.3.1 Modèle booléen 9

2.3.2 Modèle vectoriel 9

2.3.3 Modèle probabiliste 9

2.3.4 Modèles hybrides et neuronaux 10

2.4 Évaluation des systèmes de recherche d'information 11

2.4.1 Collections de test 11

2.4.2 Mesures d'évaluation 11

2.5 Limites de la recherche d'information classique 13

3 la Recherche d'Information Personnalisée. 13

3.1 modélisation de l'utilisateur 14

3.2 Pertinence utilisateur et pertinence contextuelle 14

3.3 Architecture fonctionnelle d'un SRIP 15

3.4 Profil utilisateur 16

3.5 Sources des données de profil 17

3.6 Acquisition des données 17

3.6.1 Acquisition explicite 18
3.6.2 Acquisition implicite 18
3.7 Techniques de construction des centres d'intérêts 19
3.8 Représentation du profil utilisateur 19
3.8.1 Représentation basée sur l'historique 20
3.8.2 Représentation ensembliste 20
3.8.3 Représentation connexionniste 20
3.8.4 Représentation multidimensionnelle 20
3.8.5 Représentation sémantique (conceptuelle) 20
3.9 Évolution du profil utilisateur 21
3.10 Dimensions du contexte de recherche 22
3.10.1 Dimension temporelle 22
3.10.2 Dimension spatiale / localisable 22
3.10.3 Dimension sociale 23
3.10.4 Dimension sémantique et cognitive 23
3.11 Intégration du profil utilisateur et du contexte 24
3.11.1 Reformulation personnalisée de la requête 24
3.11.2 Reclassement des résultats 24
3.12 Évaluation des systèmes de recherche d'information personnalisée 25
3.13 Limites de l'évaluation classique 25
3.14 Protocole d'évaluation 25
3.14.1 Protocole d'évaluation personnalisée 26
3.15 Mesures d'évaluation personnalisées 26
4 Conclusion 28

Chapitre II : Large Language Models (LLMs) 29

1 Introduction 30
2 Définition des LLMs 30
3 Évolution des modèles de langage 32
4 Architecture Transformer et mécanismes fondamentaux 33
5 Processus d'entraînement des LLMs 34
6 Capacités des LLMs : in-context learning et raisonnement 36
7 Agents LLM 36
7.1 Paradigmes de raisonnement agentique 38
7.2 Domaines d'application 38
7.3 Systèmes multi-agents 39
8 Agents LLM dans la recherche d'information 39
8.1 Limites de la recherche d'information classique adressées par les agents LLM 40
8.2 Architecture d'un système de RI augmenté par agents LLM 40
8.3 Retrieval itératif et recherche multi-étapes 41
8.4 Personnalisation agentique de la recherche 42

8.4.1 Intégration du profil dans le processus agentique 42
8.4.2 Mise à jour dynamique du profil par l'agent 42
8.5 Navigation web agentique pour la RI ouverte 42
8.6 Évaluation des systèmes de RI agentiques 43
8.7 Défis ouverts et perspectives 44
9 Retrieval-Augmented Generation (RAG) 44
9.1 Principe et motivation 44
9.2 Architecture d'un pipeline RAG 45
9.3 Variantes avancées : RAG modulaire et RAG agentique 45
10 Conclusion 46
Chapitre III : Conception du Système RIP-LLM **47**
1 Introduction 48
2 Profil Utilisateur 48
2.1 Partie statique 49
2.2 Partie dynamique 49
3 Processus de Recherche 53
3.1 Préparation de l'historique récent (Agent-LLM1) 53
3.2 Reformulation de la requête (LLM2) 54
3.3 Mise à jour de profil Agent-LLM3 57
4 Environnement de développement 59
4.1 *Mouter de Recherche (Tavily)* 59
4.2 `Visual Studio Code VS` 59
4.3 `API Gemini` 60
4.4 `Python` 60
4.5 `HTML` / `CSS` 60
4.6 `MySQL` 60
4.7 `XAMPP` 60
5 Interfaces du système 61
5.1 Interface d'authentification 61
5.2 Interface de recherche 62
5.3 Affichage des résultats 62
5.4 Consultation du profil utilisateur 64
6 Évaluation 65
6.1 Évaluation difficile dans le contexte de la recherche personnalisée 65
6.2 Stratégie d'Évaluation 65
6.3 Expérimentation 66
6.3.1 Résultats et discussion 69
6.4 Exemples de processus de reformulation 70
6.5 Résultats globaux de précision 73

6.5.1 Structure de la base de données 74
6.5.2 Création des tables de la base de données..... 75
6.6 Liste des tables de la base de données 76
Conclusion du Chapitre III 81
Conclusion générale **83**
Références bibliographiques **86**

Liste des figures

I.1 Architecture fonctionnelle d'un Système de Recherche d'Information Personnalisé (SRIP) d'après [1] 16
I.2 Le scénario de recherche d'informations [2] 17

II.1 Évolution des modelés GPT d'OpenAI. 31
II.2 Famille des modèles LLaMA. 32
II.3 The Transformer - model architecture. 34
II.4 Un diagramme illustre les trois étapes de notre méthode : (1) ajustement fin supervisé (SFT), (2) entraînement du modèle de récompense (RM), et (3) apprentissage par renforcement via l'optimisation de politique proximale (PPO) sur ce modèle de récompense. Les flèches bleues indiquent que ces données sont utilisées pour entraîner l'un de nos modèles. À l'étape 2, les cases A à D représentent des exemples de nos modèles classés par les étiqueteurs.[3] 35
II.5 Présentation de SELF-RAG. SELF-RAG apprend à extraire, analyser et générer des passages de texte afin d'améliorer la qualité globale, la véracité et la vérifiabilité des textes générés.[4] 45

III.1 Architecture de données : Diagramme de classes UML 49
III.2 Historique de Recherche Pertinent de l'Utilisateur 50
III.3 Prompt de AGENT-LLM1 53
III.4 AGENT-LLM1 : Prépare le contexte utilisateur a partir de son profil 54
III.5 Prompt de AGENT-LLM2(cas :generale) 55
III.6 Prompt de AGENT-LLM2 56
III.7 AGENT-LLM2 : Reformule de la requête utilisateur 57
III.8 Prompt de AGENT-LLM3 58
III.9 AGENT-LLM3 : Mis a jour du profil d'utilisateur 59
III.10 Interface du système 61
III.11 Interface de login 62
III.12 Saisie de la requete 62
III.13 les reformulation de la requete 63
III.14 les reformulation de la requete 64
III.15 Interface de visualisation du profil sémantique implicite 64
III.16 Représentation des résultats bruts (Scénario A) pour la requête « nature » de l'utilisateur 6. 72
III.17 Résultats personnalisés obtenus via l'approche RIP-LLM pour l'utilisateur 6 après reformulation. 73
III.18 la base de données 74
III.19 Les tables de la base de données. 76
III.20 Exemple de la table « Users ». 76
III.21 Exemple de la table « Requetes ». 77

III.22Exemple de la table « Termes ». 78
III.23Exemple de la table « Documents ». 78
III.24Exemple de la table « Compose »..... 79
III.25Exemple de la table « Figure ». 80
III.26Exemple de la table « Pertinence ». 81

Liste des tableaux

I.1 Principales mesures d'évaluation en Recherche d'Information..... 12
I.2 Potentially observable behaviors [5] 18

II.1 Limites RI classique vs solutions agentiques 40

III.1 Profils utilisateurs, termes extraits et requêtes reformulées (RIP-LLM)..... 67
III.2 Comparaison des performances (P@5, P@10, P@15) entre la recherche classique (A) et personnalisée RIP-LLM (B) 69
III.3 Exemples de processus de reformulation personnalisée (RIP-LLM) selon les deux scénarios 71
III.4 Synthèse globale des précisions moyennes ($P@K$) selon les deux scénarios. 74

Liste des Sigles et Acronymes

Sigle	Signification
IA	Intelligence Artificielle
LLMs	Grands Modèles de Langage
RI	Recherche d'Information
RIP	Recherche d'Information Personnalisée
SRI	Système de Recherche d'Information
NLP	Natural Language Processing
ML	Machine Learning
DL	Deep Learning
BERT	Bidirectional Encoder Representations from Transformers
GPT	Generative Pre-trained Transformer
TF	Fréquence du Terme
IDF	Fréquence Inverse des Documents
TF-IDF	Fréquence du Terme – Fréquence Inverse des Documents
BM25	Best Match 25
ODP	Open Directory Project
RAG	Retrieval-Augmented Generation
SRIP	Système de Recherche d'Information Personnalisé
VSM	Vector Space Model
LSI	Latent Semantic Indexing
SVD	Singular Value Decomposition
DPR	Dense Passage Retrieval
HyDE	Hypothetical Document Embeddings
RRF	Reciprocal Rank Fusion
TALN	Traitement Automatique du Langage Naturel
MAP	Mean Average Precision
MRR	Mean Reciprocal Rank
DCG	Discounted Cumulative Gain

nDCG	Normalized Discounted Cumulative Gain
RAGAS	Retrieval-Augmented Generation Assessment
SFT	Supervised Fine-Tuning
RLHF	Reinforcement Learning from Human Feedback
DPO	Direct Preference Optimization
PEFT	Parameter-Efficient Fine-Tuning
API	Application Programming Interface
FAISS	Facebook AI Similarity Search
BGE	BAAI General Embedding
HTTP	Hypertext Transfer Protocol
REST	Representational State Transfer
RGPD	Règlement Général sur la Protection des Données

INTRODUCTION GÉNÉRALE

INTRODUCTION GÉNÉRALE

Contexte et problématique

La recherche d'information (RI) désigne la branche de l'informatique qui porte sur l'acquisition, l'organisation, le stockage et la recherche de l'information. L'objectif de la RI est de faciliter l'accès à l'information pertinente pour un utilisateur ayant un besoin en information. Ce besoin est souvent exprimé en langage naturel par une requête composée d'un ensemble de mots-clés. La requête de l'utilisateur est soumise à un système de recherche d'information (SRI) qui permet de retrouver, à partir d'une collection de documents, un sous-ensemble de documents susceptibles d'être pertinents.

Les approches classiques de recherche d'information renvoient, en réponse à une requête utilisateur, un très grand nombre de résultats qui semblent naïvement correspondre à cette requête, même si ces résultats ne sont pas tous importants selon les intentions réelles de l'utilisateur. D'où le besoin d'avoir des connaissances contextuelles sur l'utilisateur pour pouvoir adapter les résultats de recherche à ses intentions, et par conséquent améliorer la pertinence des résultats ; c'est ainsi que la recherche d'information personnalisée (RIP) est apparue. L'idée principale de la RIP est qu'une même réponse pour une même requête envoyée par deux utilisateurs différents peut ne pas les satisfaire tous deux. Ainsi, l'objectif des systèmes de recherche d'information personnalisée (SRIP) est d'inférer le réel besoin caché derrière la requête afin de guider le processus de recherche vers les résultats les mieux correspondants.

Pour filtrer les requêtes ambiguës et classer les résultats les plus pertinents selon les intentions de l'utilisateur, les approches de RIP ont besoin de prendre en compte l'utilisateur et les données le caractérisant au cœur du processus. À cet effet, les recherches se sont principalement concentrées sur la modélisation de l'utilisateur. La modélisation de l'utilisateur est le processus d'acquisition, d'extraction et de représentation des données caractérisant ses centres d'intérêts, ses besoins cognitifs ou ses préférences. Cette structure est appelée dans la littérature « profil d'utilisateur ». Ce profil regroupe les données que le système perçoit sur un utilisateur donné afin de l'intégrer, par la suite, dans le processus de recherche.

L'hypothèse de base des approches de RIP est que les nouvelles recherches de l'utilisateur sont souvent en relation avec ses interactions précédentes. De ce fait, les concepteurs des profils ne se limitent pas aux données statiques explicitement exprimées (identité, thèmes déclarés), mais accordent plus d'importance aux données contextuelles dynamiques. Pour améliorer la représentativité du profil, la littérature exploite différentes dimensions contextuelles : la dimension sociale (communautés et interactions), la dimension localisable (emplacement géographique), la dimension temporelle (événements et chronologie), et la dimension sémantique (signification véhiculée par les termes).

Parallèlement, le domaine de l'intelligence artificielle a connu une rupture paradigmatique avec l'émergence des Grands Modèles de Langage (Large Language Models – LLMs), tels que GPT, Llama ou Gemini. Grâce à leurs capacités de compréhension sémantique profonde et de raisonnement textuel, ces modèles ouvrent une nouvelle ère pour la RIP. Ils permettent d'améliorer la compréhension et la reformulation des requêtes utilisateur en exploitant le contexte et le profil de l'utilisateur, et ce, spécifiquement en amont du processus de recherche.

Les principales contributions

D'après la recherche bibliographique que nous avons menée, nous avons constaté que :

— Les approches de recherche d'information centrées utilisateur traitent et exploitent souvent

les différents aspects contextuels (dimensions thématiques, temporelles et comportementales) indépendamment l'un de l'autre.

— Les systèmes existants peinent à unifier ces dimensions dans un cadre dynamique unique pour lever efficacement l'ambiguïté des requêtes courtes et vagues.

Nous estimons que l'exploitation conjointe, guidée par les capacités sémantiques des Grands Modèles de Langage, des différentes dimensions contextuelles de l'utilisateur (historique récent et profil persistant) est d'une grande utilité pour décrire la per- tinence utilisateur et lever l'ambiguïté des requêtes. Les principales contributions de ce mémoire, concrétisées par le développement du système *RIP-LLM*, se résument dans les trois points suivants :

1. Proposer une approche de modélisation basée sur un profil utilisateur , combinant une dimension thématique à long terme (profil persistant : lieu de résidence, spécialité) et une dimension comportementale à court terme issue de l'analyse des interactions récentes combinée à une représentation statistique des termes (type TF-IDF).
2. Développer un mécanisme d'identification du contexte de recherche basé sur les LLMs, capable d'analyser l'historique des requêtes récentes conjointement avec les termes pondérés afin de cerner l'intention réelle et lever l'ambiguïté de la requête actuelle.
3. Concevoir un système de reformulation et d'enrichissement sémantique personnalisé, exploitant un second LLM pour générer quatre à cinq alternatives de requêtes contextualisées. Ce module permet à l'utilisateur de choisir la formulation idéale ou de conserver sa requête originale afin d'éviter l'effet de bulle de filtres.

L'objectif de notre approche est d'exploiter les informations fournies par ce couplage entre profil et LLM pour améliorer une phase cruciale et déterminante du processus de recherche : l'enrichissement et la reformulation de la requête, afin d'améliorer la formulation du besoin informationnel et d'augmenter indirectement la pertinence des résultats.

Plan du mémoire

Le mémoire est organisé en trois chapitres principaux, suivis d'une conclusion et des perspectives :

— **Chapitre 1 :** Il est réservé à l'étude de la recherche d'information classique et aux fondements de la personnalisation. Nous y mettons en évidence le processus de RI, les modèles formels, les techniques d'évaluation, ainsi que les concepts clés de la RIP (modélisation, profils et dimensions contextuelles).

— **Chapitre 2 :** Il est dédié aux Grands Modèles de Langage (LLMs). Ce chapitre met en lumière l'évolution de leurs architectures, leurs paradigmes d'utilisation (Prompt Engineering, agents intelligents) ainsi que leur rôle dans les tâches de réécriture et d'expansion de requêtes.

— **Chapitre 3 :** Il présente le cœur de notre contribution à travers la conception, la réalisation technique et l'évaluation expérimentale du système proposé RIP-LLM. Ce chapitre expose l'architecture des modules (Profilage, Contexte, Reformulation), les choix d'implémentation et l'analyse des performances obtenues.

Chapitre I :

De la recherche d'information Classique à la personnalisation

1 Introduction

La capacité à accéder rapidement et efficacement à l'information pertinente constitue l'un des défis majeurs de notre ère numérique. Chaque jour, des milliards d'utilisateurs interagissent avec des systèmes de recherche pour satisfaire des besoins informationnels extrêmement variés : un étudiant préparant un exposé, un médecin cherchant les dernières avancées thérapeutiques, un ingénieur consultant la documentation technique, ou un simple curieux explorant un domaine nouveau. Face à cette diversité des besoins et à l'explosion du volume de données disponibles, les systèmes capables de fournir des réponses pertinentes, adaptées et contextualisées sont devenus indispensables [6].

La **Recherche d'Information (RI)** est la discipline scientifique qui étudie les méthodes, modèles et systèmes permettant de retrouver des documents ou des informations pertinentes au sein de grandes collections, en réponse à un besoin exprimé par un utilisateur. Née dans les années 1960 avec les premiers travaux de Gerard Salton sur les systèmes d'indexation automatique, la RI a traversé plusieurs révolutions technologiques et conceptuelles pour devenir aujourd'hui l'infrastructure fondamentale des moteurs de recherche, des bibliothèques numériques, des systèmes de recommandation, et des assistants intelligents [7].

Dans sa forme **classique**, la RI repose sur des modèles formels "booléen, vectoriel, probabiliste "qui traitent la recherche comme un problème de correspondance entre représentations de requêtes et représentations de documents. Ces approches, remarquablement efficaces pour la recherche à grande échelle, ont cependant une limitation fondamentale : elles ignorent l'individu. Elles supposent implicitement qu'une même requête exprime le même besoin pour tous les utilisateurs, et que les résultats pertinents pour l'un le seront également pour l'autre [8]. Or, l'expérience quotidienne nous enseigne le contraire : la requête "Python" n'a pas le même sens pour un développeur informatique, un biologiste spécialiste des reptiles, et un enfant fasciné par les serpents.

C'est pour répondre à cette inadéquation fondamentale qu'a émergé la **Recherche d'Information Personnalisée (RIP)**. En intégrant des informations sur l'utilisateur :ses intérêts, son expertise, son historique, ses préférences et sur son contexte de recherche ,sa localisation, le moment de la journée, son dispositif, sa tâche en cours , les systèmes personnalisés visent à fournir des réponses véritablement adaptées à chaque individu dans chaque situation [9].

Ce chapitre pose les fondements théoriques et pratiques nécessaires à la compréhension de ces deux paradigmes. Dans un premier temps, nous présentons les concepts et mécanismes de la recherche d'information classique : ses notions de base, son processus, ses modèles formels, ses méthodes d'évaluation et ses limites. Dans un second temps, nous exposons les approches de modélisation de l'utilisateur qui constituent le cœur de la personnalisation : comment capturer, représenter et exploiter le profil d'un utilisateur pour améliorer la pertinence des résultats. Nous abordons ensuite les dimensions contextuelles qui enrichissent cette personnalisation, avant de décrire les mécanismes d'intégration du profil dans le processus de recherche et les méthodes d'évaluation spécifiques aux systèmes personnalisés [10, 11].

2 Recherche d'Information Classique

La recherche d'information classique représente l'ensemble des approches, modèles et techniques développés depuis les années 1960 pour permettre l'accès automatique à des documents pertinents au sein de collections textuelles. Ces approches constituent encore aujourd'hui le socle sur lequel sont construits la plupart des systèmes de recherche modernes, y compris les moteurs

de recherche commerciaux comme Google ou Bing [12].

La caractéristique fondamentale de la RI classique est son approche **orientée document et indépendante de l'utilisateur** : la pertinence est définie comme une propriété de la relation entre le document et la requête, sans considération du contexte, des connaissances préalables, ou des objectifs spécifiques de l'utilisateur qui a formulé cette requête. Cette abstraction, bien qu'une simplification de la réalité, a permis le développement de systèmes scalables capables de traiter des milliards de documents en quelques millisecondes [7].

2.1 Notions de base de la recherche d'information

La recherche d'information repose sur un ensemble cohérent de notions fondamentales qui structurent le fonctionnement des systèmes de recherche d'information (SRI). Ces notions forment le cadre conceptuel permettant de modéliser l'interaction entre l'utilisateur, son besoin informationnel, les documents disponibles, et les mécanismes automatiques de recherche, de correspondance et de classement. Elles sont indispensables pour comprendre à la fois l'architecture des SRI et les principes qui régissent leur performance et leur évaluation [7].

Dans ce contexte, l'information brute contenue dans les documents n'est pas exploitée directement par le système. Elle subit une transformation vers une représentation intermédiaire abstraite à travers un processus clé appelé **indexation**. Cette opération consiste à analyser le contenu documentaire afin d'en extraire des unités significatives telles que les termes, concepts ou descripteurs puis à leur associer des poids numériques reflétant leur importance relative. Ces pondérations visent à capturer, de manière approximative mais efficace, la contribution sémantique de chaque unité au contenu global du document][6].

2.1.1 Terme, document, requête et collection

Les concepts de **terme**, **document**, **requête** et **collection** constituent les éléments fondamentaux sur lesquels repose tout système de recherche d'information. Ils permettent de formaliser la manière dont l'information est représentée, interrogée et organisée au sein d'un SRI [12].

Le **terme** correspond à une unité linguistique extraite du contenu textuel lors du processus d'indexation. Il peut s'agir d'un mot, d'un radical (stem), d'un lemme ou d'une expression composée (n-gramme). Les termes servent de base à la représentation des documents et des requêtes et sont généralement associés à des poids numériques reflétant leur importance relative dans le contexte documentaire. Cette pondération vise à capturer la contribution sémantique de chaque terme au contenu global du document, comme dans les modèles TF-IDF (Term Frequency - Inverse Document Frequency) ou BM25 (Best Match 25), qui sont largement utilisés dans les systèmes modernes [7].

Le **document** désigne toute entité informationnelle susceptible de répondre à un besoin d'information. Dans le cadre de la recherche d'information, un document peut prendre diverses formes : un texte, une page Web, un article scientifique, un rapport technique, une image accompagnée de métadonnées, ou même un fichier multimédia. Afin de faciliter la recherche automatique, le document est transformé en une représentation abstraite fondée sur les termes qui le caractérisent, ce qui permet sa comparaison avec les requêtes des utilisateurs selon des mesures de similarité appropriées [7].

La **requête** est l'expression explicite du besoin informationnel formulée par l'utilisateur. Elle est souvent concise, parfois ambiguë et fréquemment incomplète, ce qui peut engendrer une divergence entre l'intention réelle de l'utilisateur et la représentation traitée par le système. Cette difficulté, connue sous le nom de **fossé d'expression** ou **vocabulary mismatch**, constitue un défi

majeur pour les systèmes de recherche d'information, car elle limite leur capacité à interpréter correctement les besoins sous-jacents][13].

La **collection** correspond à l'ensemble des documents indexés par le système et disponibles pour la recherche. Sa taille, son hétérogénéité thématique, sa langue et son domaine d'application influencent directement les stratégies d'indexation adoptées, les modèles de recherche utilisés et les performances globales du système. Par exemple, une collection homogène spécialisée dans un domaine précis (médecine, droit) nécessitera des techniques d'indexation différentes de celles appliquées à une collection générale et hétérogène comme le Web [6].

2.1.2 Besoin d'information et pertinence

A. Le besoin d'information :
comprend le processus cognitif qui pousse l' utilisateur à effectuer des recherches. Cela indique un manque de connaissances ou insuffisance sur un sujet particulier et n'est pas toujours clairement indiqué dans le texte. Selon [14], les besoins d' information évoluent dynamiquement tout au long du processus de recherche en fonction des résultats et de l' interaction évolutive du système . En matière de communication, on distingue trois catégories de besoins informationnels :

• **Besoin vérificatif (verificatory need)** :L' utilisateur recherche une confirmation ou une vérification de faits déjà connus . Par exemple, la recherche d' un article à partir d'une adresse précise ou de la date de publication d' un ouvrage dont la référence est connue. Ce type de besoin est considéré comme stable, c'est-à-dire qu'il ne varie pas au cours de la recherche .

• **Besoin thématique connu (known thematic need)** : L'utilisateur souhaite approfondir sa compréhension d'un domaine qu'il connaît déjà. L'utilisateur affine fréquemment sa requête au cours de la recherche ; ce besoin peut être constant ou fluctuant. Il s'exprime également de manière incomplète en ne déclarant qu'une partie de ce qu'il sait réellement .

• **Besoin thématique inconnu (unknown thematic / exploratory need)** :L'utilisateur explore un nouveau domaine et recherche des points de vue ou des liens inédits en dehors des sujets qui lui sont familiers . Ce type de besoin est par nature imprévisible et s'exprime de manière inachevée , reflétant l'incertitude et la découverte progressive .

B. La pertinence :
désigne le degré d'adéquation entre un document restitué par le système et le besoin informationnel réel de l'utilisateur. Il s'agit d'un concept central mais intrinsèquement complexe et multidimensionnel, car il dépend non seulement du contenu du document, mais également du contexte d'utilisation, des objectifs poursuivis, et des connaissances préalables de l'utilisateur.[8] propose une classification détaillée des dimensions de la pertinence :

• **La pertinence algorithmique** : Mesurée par le système sur la base de la correspondance formelle entre la requête et le document, calculée via des scores de similarité. C'est le seul type de pertinence indépendant du contexte utilisateur.

• **La pertinence thématique** : Évalue si le document traite effectivement du sujet recherché. C'est le type de pertinence adressé par les assesseurs des campagnes d'évaluation comme TREC.

• **La pertinence cognitive** : Prend en compte l'apport du document aux connaissances de l'utilisateur et sa capacité à réduire son incertitude informationnelle. Elle est "pondérée" par la perception ou les connaissances de l'utilisateur sur le thème de la requête.

• **La pertinence situationnelle** : Considère l'utilité concrète du document dans le contexte spécifique de l'utilisateur, incluant ses contraintes temporelles, ses objectifs immédiats et son environnement de travail. Elle traduit essentiellement l'utilité de l'information relativement au but de recherche de l'utilisateur.

La **RI personnalisée** explore essentiellement la pertinence cognitive et la pertinence situationnelle, là où la RI classique se limite principalement à la pertinence algorithmique et thématique

[13, 8].

2.1.3 Système de Recherche d'Information (SRI)

Un **Système de Recherche d'Information (SRI)** est un ensemble organisé de composants logiciels, d'algorithmes et de structures de données dont l'objectif est de permettre à un utilisateur d'accéder, de manière efficace et pertinente, aux documents les mieux adaptés à son besoin informationnel parmi une collection souvent volumineuse et hétérogène. Le SRI agit comme un intermédiaire intelligent entre l'utilisateur et la masse documentaire, en assurant les fonctions essentielles de représentation, d'indexation, de recherche, de correspondance et de classement des documents [6].

2.2 Processus de recherche d'information

Le processus de recherche d'information décrit l'ensemble des étapes séquentielles et interconnectées permettant de transformer un besoin informationnel exprimé par l'utilisateur en une liste ordonnée de documents pertinents. Ce processus constitue le cœur opérationnel des systèmes de recherche d'information et conditionne directement leur efficacité, leur rapidité et la satisfaction des utilisateurs [7].

2.2.1 Expression du besoin en information

L'**expression du besoin en information** constitue le point de départ du processus de recherche. À cette étape, l'utilisateur doit traduire un besoin cognitif interne, souvent flou et évolutif, en une requête formelle que le système peut interpréter et traiter. Cette traduction n'est pas triviale : elle nécessite que l'utilisateur sélectionne des termes appropriés, anticipe le vocabulaire utilisé dans les documents pertinents, et structure sa requête de manière compréhensible pour le système [14].

Plusieurs difficultés apparaissent à cette étape. D'une part, l'utilisateur peut ne pas disposer du vocabulaire technique ou spécialisé nécessaire pour exprimer précisément son besoin, surtout s'il explore un domaine qui lui est peu familier. D'autre part, une requête trop générale génère un grand nombre de résultats peu pertinents (**bruit informationnel**), tandis qu'une requête trop spécifique peut exclure des documents pertinents (**silence informationnel**). Cette étape est donc critique pour la qualité globale des résultats, car une formulation inadéquate ou incomplète peut entraîner des résultats décevants, même si le système dispose d'excellents algorithmes de recherche [8].

2.2.2 Processus d'indexation

L'**indexation** est le processus central qui transforme le contenu brut des documents en une représentation structurée, compacte et exploitable par les algorithmes de recherche. Sans indexation efficace, un système de recherche d'information serait incapable de traiter rapidement des collections de grande taille. L'indexation vise un double objectif : réduire l'espace de stockage nécessaire et accélérer les opérations de recherche, tout en préservant la capacité du système à discriminer entre documents pertinents et non pertinents [6].

L'**analyse lexicale** (ou tokenization) :
L'étape qui consiste à transformer le texte d'un document en une collection de termes est connue

sous le nom d'analyse lexicale. Un terme fait référence à une unité ou entité lexicale, également

connu sous le nom de Token en anglais. L'analyseur lexical a la capacité d'identifier les espaces de séparation des mots, les chiffres, la ponctuation, etc., et de les éliminer.

L'**Élimination des mots vides** (stop words) :
L'exclusion des mots vides contribue à rendre le texte à indexer plus simple et plus léger. Un mot inintéressant est un terme dénué de sens dans un document, une expression qui se manifeste avec une occurrence similaire à travers l'ensemble des écrits, donc un terme qui ne facilite pas la différenciation entre les divers textes. Un terme vide [15] peut être un pronom personnel, une préposition ou tout autre mot grammatical. De plus, il peut s'agir d'un mot athématique, signifiant qu'il peut apparaître dans n'importe quel document puisqu'il présente le sujet sans le traiter (par exemple : contenir, appartenir...).

La **Normalisation** :
vise à harmoniser les différentes formes lexicales d'un même concept afin de regrouper les termes sémantiquement proches sous une représentation unique. Cette étape est essentielle pour améliorer le rappel du système, c'est-à-dire sa capacité à retrouver tous les documents pertinents, même s'ils utilisent des variantes morphologiques ou orthographiques des termes de la requête.

Deux approches principales sont utilisées :

1. **Le stemming (racinisation)** : Technique consistant à réduire les mots à leur racine ou radical en supprimant les suffixes, préfixes et désinences. Par exemple, "chercher", "chercheur", "recherche" et "cherchant" sont tous réduits à la racine "cherch". Les algorithmes les plus connus sont l'algorithme de Porter pour l'anglais et l'algorithme de Snowball pour plusieurs langues. Le stemming est rapide et efficace, mais peut parfois produire des racines non lexicales.

2. **La lemmatisation** : Approche plus sophistiquée consistant à ramener chaque mot à sa forme canonique ou lemme (infinitif pour les verbes, singulier masculin pour les noms et adjectifs). Par exemple, "suis", "sommes", "était" sont tous lemmatisés en "être". La lemmatisation nécessite une analyse morphologique et syntaxique plus approfondie, mais produit des résultats plus précis et linguistiquement corrects [6].

Pondération des termes (TF, IDF) :

La **pondération des termes** constitue l'étape finale et cruciale de l'indexation. Elle consiste à attribuer à chaque terme présent dans un document un poids numérique reflétant son importance relative pour caractériser le contenu de ce document. Les schémas de pondération les plus utilisés combinent deux facteurs complémentaires [7] :

1. TF «Term Frequency» (Fréqeunced'untermedansledocument) : Mesure la fréquence du terme dans le document, c'est-à-dire le nombre d'occurrences du terme dans le document (pondération locale). Un terme apparaissant fréquemment dans un document est généralement plus représentatif de son contenu.

Il existe plusieurs formules pour calculer TF, les plus connues sont :

$$tf_{ij} = freq_{ij}$$

$$tf_{ij} = 1 + \log(freq_{ij})$$

$$tf_{ij} = \frac{freq_{ij}}{\max(freq_j)}$$

$$tf_{ij} = \frac{freq_{ij}}{K + freq_{ij}}$$

K introduit pour tenir compte de la longueur des documents

$$tf_{ij} = \frac{freq_{ij}}{freq_{ij} + 0.5 + 1.5 \times \frac{longueur_doc}{longueur_moy_doc}}$$

2. IDF «inverse document frequency » (Fréquence inverse du document) :

Mesure la fréquence inverse du terme dans la collection (pondération globale). Un terme apparaissant dans de nombreux documents est moins discriminant qu'un terme rare. Il est calculé comme :

Il ya deux formules pour calculer l'IDF d'un terme ti :

$$IDF_i = \log\left(\frac{N}{n_i}\right) \qquad IDF_i = \log\left(\frac{N}{n_i} + 1\right)$$

Avec :

- N : le nombre total de documents de la collection ;
- n_i : le nombre de documents contenant le terme t_i ;
- df_t : le nombre de documents contenant le terme t.

3. TF-IDF

Le schéma **TF-IDF** combine ces deux facteurs : un terme reçoit un poids élevé s'il est fréquent dans le document mais rare dans la collection. Ce schéma simple mais efficace a dominé la recherche d'information pendant plusieurs décennies [7].

$$TF \times IDF = \log(1 + TF) \times \log\left(\frac{N}{dft}\right)$$

4. BM25

Le modèle **BM25** (Best Match 25), développé par Robertson et ses collaborateurs (1994), améliore TF-IDF en introduisant des paramètres de saturation et une normalisation par la longueur des documents :

$$BM25(D, Q) = \sum_{t \in Q} IDF(t) \cdot \frac{f(t, D) \cdot (k_1 + 1)}{f(t, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{avgdl}\right)}$$

Avec :

- q_i : un terme de la requête
- $TF(q_i)$: fréquence du terme dans le document
- $IDF(q_i)$: importance du terme dans le corpus
- $|D|$: longueur du document
- $avgDL$: longueur moyenne des documents
- k_1 : paramètre de saturation (souvent $\approx$ 1.2 à 2)
- b : paramètre de normalisation de la longueur (souvent $\approx$ 0.75)

BM25 aujourd'hui est considéré comme l'un des modèles de pondération les lus performants et est largement utilisé dans les moteurs de recherche modernes [12].

2.2.3 Reformulation de la requête

La **reformulation de la requête** est une technique visant à améliorer la qualité des résultats en modifiant, enrichissant ou raffinant la requête initiale formulée par l'utilisateur. Cette étape peut être effectuée manuellement par l'utilisateur, ou automatiquement par le système à l'aide de diverses techniques algorithmiques [6].

Les principales approches de reformulation automatique incluent

— l'**expansion de requête** (ajout de termes supplémentaires provenant de thésaurus, dictionnaires de synonymes ou co-occurrences statistiques).

— le **retour de pertinence** (le modèle de Rocchio où l'utilisateur indique quels documents sont pertinents pour enrichir la requête).

— la **désambiguïsation** (identification du sens voulu par l'utilisateur pour les termes polysémiques).

— la **correction orthographique** (détection et correction automatique des erreurs de frappe).

La reformulation de requête constitue un mécanisme puissant pour combler le fossé lexical entre le vocabulaire de l'utilisateur et celui des documents, mais elle doit être appliquée avec prudence pour éviter de dériver vers des résultats non pertinents " phénomène connu sous le nom de **query drift**" [12].

2.2.4 Appariement document–requête

L'**appariement document–requête** (matching) est l'opération centrale qui consiste à calculer un score de similarité ou de pertinence entre la requête traitée et chaque document indexé de la collection. Ce score détermine l'ordre dans lequel les documents seront présentés à l'utilisateur, les documents les mieux notés apparaissant en tête de liste [6].

Le calcul de similarité dépend fondamentalement du modèle de recherche d'information adopté par le système. Dans le modèle vectoriel, On a plusieurs modèles de calcule la similarité, et parmi les modelés on a le **cosinus de l'angle** entre le vecteur requête et le vecteur document :

$$\text{sim}(D, Q) = \cos(\theta) = \frac{\sum_t w_{t,D} \cdot w_{t,Q}}{\sqrt{\sum_t w_{t,D}^2} \cdot \sqrt{\sum_t w_{t,Q}^2}}$$

où $w_{t,D}$ et $w_{t,Q}$ sont les poids du terme t dans le document et dans la requête respectivement [7]. Cette mesure a l'avantage d'être normalisée (elle varie entre 0 et 1) et indépendante de la longueur des documents. Une fois les scores calculés pour tous les documents, le système effectue un classement (ranking) par ordre décroissant de score et retourne les k meilleurs résultats à l'utilisateur.

2.3 Modèles de recherche d'information

Les **modèles de recherche d'information** constituent le cadre formel et mathématique qui définit comment les documents et les requêtes sont représentés, comment la pertinence est calculée, et comment les résultats sont classés. Selon Baeza-Yates et Ribeiro-Neto (1999), un modèle de RI peut être caractérisé par un **quadruplet** $(D, Q, F, R(q, d))$ où D est l'ensemble des documents, Q l'ensemble des requêtes, F le schéma de représentation théorique des documents

et requêtes, et $R(q, d)$ la fonction de pertinence évaluant l'adéquation entre un document d et une requête q. Les modèles classiques peuvent être regroupés en trois grandes familles : booléens, vectoriels et probabilistes [7].

2.3.1 Modèle booléen

Le **modèle booléen** [16] est historiquement le premier modèle formel de recherche d'information. Il est fondé sur la **théorie des ensembles** et la **logique booléenne**, utilisant les opérateurs **AND** (intersection), **OR** (union), et **NOT** (négation).

Dans ce modèle, chaque document est représenté par une conjonction logique des termes non pondérés qui constituent son index :

$$d_j = t_1 \wedge t_2 \wedge t_3 \wedge \cdots \wedge t_n \tag{2.1}$$

Une requête est une expression booléenne combinant des termes avec les opérateurs logiques. Par exemple :

$$q = (t_1 \wedge t_2) \vee (t_3 \wedge t_4) \tag{2.2}$$

La fonction de correspondance est **binaire** : un document correspond à la requête ou ne correspond pas :

$$RSV(q, d) = \{1, 0\} \tag{2.3}$$

2.3.2 Modèle vectoriel

Le **modèle vectoriel** (Vector Space Model, VSM), proposé par Salton dans les années 1970, représente une avancée majeure par rapport au modèle booléen. Il introduit la notion de **pertinence partielle** et de **classement des résultats** [16].

Documents et requêtes sont représentés comme des vecteurs dans un espace vectoriel multidimensionnel à N dimensions (une par terme d'indexation) :

- **Vecteur document** : $\vec{D}_j = (d_{1j}, d_{2j}, d_{3j}, ..., d_{nj})$ (2.4)

- **Vecteur requête** : $\vec{Q} = (q_1, q_2, q_3, ..., q_n)$ (2.5)

Il ya plusieurs mesures de similarité ont été définies, les principales étant :

- **Produit scalaire** : $SIM(Q, D_j) = \sum_{i=1}^{N} q_i \cdot d_{ij}$ (2.6)

- **Mesure de Jaccard** : $SIM(Q, D_j) = \frac{\sum_{i=1}^{N} q_i \cdot d_{ij}}{\sum_{i=1}^{N} q_i^2 + \sum_{i=1}^{N} d_{ij}^2 - \sum_{i=1}^{N} q_i \cdot d_{ij}}$ (2.7)

- **Mesure cosinus** : $SIM(Q, D_j) = \frac{\sum_{i=1}^{N} q_i \cdot d_{ij}}{\sqrt{\sum_{i=1}^{N} q_i^2} \cdot \sqrt{\sum_{i=1}^{N} d_{ij}^2}}$ (2.8)

2.3.3 Modèle probabiliste

Le **modèle probabiliste** [17] aborde la recherche d'information sous l'angle de la théorie des probabilités. Il vise à estimer la **probabilité qu'un document soit pertinent** pour une requête

donnée, en s'appuyant sur le **Probability Ranking Principle (PRP)** qui stipule que le classement optimal consiste à ordonner les documents par probabilité décroissante de pertinence.

Documents et requêtes sont représentés par des vecteurs booléens dans un espace à n dimensions :

- $D_j = (w_{1,j}, w_{2,j}, w_{3,j}, ..., w_{n,j})$ (2.9)
- $Q = (w_{1,q}, w_{2,q}, w_{3,q}, ..., w_{n,q})$, avec $w_{k,j} \in \{0, 1\}$ (2.10)

Le score d'appariement RSV entre le document D et la requête Q est donné par le rapport de vraisemblance :

$$RSV(Q, D) = \frac{P(R|D)}{P(\bar{R}|D)} \tag{2.11}$$

Le modèle probabiliste a donné lieu à de nombreuses extensions, dont le système **Okapi BM25** :

$$w_{td} = \log\left(\frac{N - df + 0.5}{df + 0.5}\right) \times \frac{(K_1 + 1) \times TF}{K_1 \times \left((1 - b) + b \times \frac{dl}{avgdl}\right) + tf} \tag{2.12}$$

où $dl = moy_{d_i \in D}(dl_i)$ et K_1, b sont des paramètres dépendant de la collection.

Avantages : Fondements théoriques solides, gestion de la saturation de fréquence, normalisation par longueur de document, performances état de l'art.

Limites : Complexité computationnelle, nécessité d'ajustement de paramètres, hypothèse d'indépendance des termes [12].

2.3.4 Modèles hybrides et neuronaux

Les **modèles hybrides** combinent plusieurs approches de recherche d'information pour exploiter leurs forces respectives. Les **modèles neuronaux** représentent la frontière actuelle de la recherche en RI. Ils utilisent des réseaux de neurones profonds pour apprendre des représentations vectorielles denses (embeddings) capturant des relations sémantiques complexes que les modèles classiques ne peuvent pas représenter [7].

Parmi les approches notables, on distingue :

le **modèle connexionniste**, dont les SRI basés sur cette approche utilisent le fondement des réseaux de neurones pour la modélisation des unités textuelles et la mise en œuvre du processus de RI. L'idée de base est que la RI est un processus associatif représentable par les mécanismes de propagation d'activation des réseaux de neurones.

Deux architectures principales ont été explorées :

1. les **modèles à auto-organisation** : (basés sur les cartes de Kohonen) .
2. les **modèles à couches** (comme MERCURE), composés de trois couches interconnectées :couche requête (Q), couche termes (T) et couche documents (D).

Le **modèle LSI** (Latent Semantic Indexing) exploite la coexistence des termes et réduit l'espace des termes via la décomposition en valeurs singulières (SVD), en regroupant les termes communs dans les mêmes dimensions pour obtenir des représentations dans un espace plus petit composé de concepts de niveau supérieur.

Les **modèles de représentation dense** utilisent des encodeurs neuronaux basés sur des architectures comme BERT pour projeter requêtes et documents dans un espace vectoriel dense.

Les **modèles de langage pré-entraînés** (ColBERT, DPR, ANCE) ont révolutionné le domaine en permettant l'apprentissage de représentations contextuelles à partir de corpus massifs.

Dans la pratique, les systèmes modernes utilisent souvent une architecture hybride à deux étages : BM25 pour récupérer un ensemble de candidats, puis des modèles neuronaux pour re-classer finement ces candidats (**two-stage retrieval**).

2.4 Évaluation des systèmes de recherche d'information

L'**évaluation des systèmes de recherche d'information** est une composante essentielle permettant de mesurer objectivement la qualité des résultats produits, de comparer différentes approches, et de guider les améliorations. Elle repose sur trois piliers fondamentaux : les collections de test standardisées, les mesures quantitatives de performance, et les protocoles expérimentaux rigoureux [6, 7]

2.4.1 Collections de test

Les **collections de test** sont des ensembles de données standardisés comprenant trois composants : un **corpus de documents**, un **ensemble de requêtes (topics)** avec descriptions détaillées, et des **jugements de pertinence** annotés manuellement pour chaque paire (requête, document) [13, 12].

Les collections les plus influentes incluent :

• **TREC** (Text REtrieval Conference, organisée annuellement depuis 1992 par le NIST avec diverses pistes thématiques).

• **CLEF** (focalisée sur la recherche multilingue et cross-lingue).

• **NTCIR** (orientée vers les langues asiatiques).

• **MS MARCO** (collection massive moderne avec millions de requêtes réelles issues de Bing, largement utilisée pour l'évaluation des modèles neuronaux).

2.4.2 Mesures d'évaluation

La **Précision** mesure la proportion de documents pertinents parmi tous les documents retournés :

$$\text{Précision} = \frac{\text{Documents pertinents retournés}}{\text{Total documents retournés}} = \frac{TP}{TP + FP}$$

Le **Rappel** mesure la proportion de documents pertinents retournés parmi tous les documents pertinents existants :

$$\text{Rappel} = \frac{\text{Documents pertinents retournés}}{\text{Total documents pertinents}} = \frac{TP}{TP + FN}$$

La **F-mesure** est la moyenne harmonique de la précision et du rappel :

$$\text{F-mesure} = 2 \times \frac{\text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}}$$

Il existe généralement un **compromis (trade-off)** entre précision et rappel : en retournant plus de documents, on augmente le rappel mais on risque de diminuer la précision [7].

Ces métriques intègrent la dimension du classement des résultats, essentielle pour l'utilisateur.

La **MAP (Mean Average Precision)** calcule la précision moyenne pour chaque requête en tenant compte du rang des documents pertinents :

$$AP = \frac{1}{R} \sum_{k=1}^{N} \text{Précision}(k) \times \text{rel}(k)$$

Le **MRR (Mean Reciprocal Rank)** est utile pour les tâches où seul le premier résultat pertinent importe :

$$MRR = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{\text{rang}_q}$$

Le **nDCG (normalized Discounted Cumulative Gain)** [18] gère des jugements de pertinence gradués. Le DCG au rang k est :

$$DCG@k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

Le nDCG normalise le DCG par le DCG idéal (IDCG) :

$$nDCG@k = \frac{DCG@k}{IDCG@k}$$

Le nDCG varie entre 0 et 1, où 1 représente le classement parfait. Il est devenu la mesure de référence pour l'évaluation moderne des systèmes de recherche [18, 8]

TABLE I.1 – Principales mesures d'évaluation en Recherche d'Information

Mesure	Formule	Utilité	Caractéristiques
Précision	$\frac{\text{Docs pertinents retournés}}{\text{Total docs retournés}}$	Qualité des résultats (réduction du bruit)	— Varie 0–1 — Privilégie la qualité
Rappel	$\frac{\text{Docs pertinents retournés}}{\text{Total docs pertinents}}$	Complétude de récupération	— Varie 0–1 — Privilégie l'exhaustivité
F-mesure	$2 \times \frac{Prcision \times Rappel}{Prcision + Rappel}$	Combine précision et rappel	— Moyenne harmonique — Pénalise déséquilibres
MAP	$\frac{1}{Q} \sum_{q \in Q} \frac{1}{R_q} \sum_{k} P(k) \cdot rel(k)$	Performance globale sur requêtes	— Sensible au rang — Pertinence binaire
MRR	$\frac{1}{Q} \sum_{q \in Q} \frac{1}{rang_q}$	Position du premier doc pertinent	— Très utilisée en QA — Ignore autres documents
nDCG@k	$\frac{DCG@k}{IDCG@k}$	Gain cumulé normalisé	— Multi-niveaux — Normalisée 0–1 — Mesure moderne

2.5 Limites de la recherche d'information classique

Bien que les systèmes de recherche d'information classiques aient connu un succès remarquable et continuent d'être largement utilisés, ils présentent plusieurs limites fondamentales qui ont motivé le développement d'approches plus avancées [7].

1. Dépendance lexicale et lacune sémantique : Les modèles classiques reposent sur des correspondances lexicales et ne capturent pas la sémantique profonde : synonymie, polysémie, relations conceptuelles, implications logiques. Un document pertinent utilisant un vocabulaire différent de celui de la requête risque de ne pas être retrouvé (problème du **vocabulary mismatch**)
.

2. Traitement limité de l'ambiguïté : Les requêtes courtes et ambiguës posent un problème majeur. Sans information contextuelle, le système ne peut pas déterminer l'intention réelle de l'utilisateur. Par exemple, la requête "virus" peut référer à un virus informatique ou biologique selon le profil de l'utilisateur.

3. Absence de personnalisation : Les systèmes classiques traitent toutes les requêtes identiques de la même manière, ignorant le profil, les connaissances préalables, la profession, ou les intérêts de l'utilisateur. Deux informaticiens et deux biologistes soumettant la même requête "virus" recevront exactement les mêmes résultats, alors que leurs besoins sont radicalement différents [14].

4. Hypothèse d'indépendance des termes : La plupart des modèles classiques supposent que les termes sont indépendants les uns des autres (bag-of-words assumption), ignorant les relations syntaxiques, les dépendances contextuelles, et la structure des phrases [12].

5. Surcharge informationnelle : Avec l'explosion du volume de données, les systèmes retournent souvent des milliers de résultats. Les utilisateurs sont submergés et peinent à filtrer l'information de qualité. De plus, les modèles classiques ne jugent pas intrinsèquement la crédibilité, l'autorité ou la fraîcheur des sources [19].

6. Volume, hétérogénéité et disparité : Le volume d'information se mesure en téraoctets voire en pétaoctets, posant des problèmes de **passage à l'échelle**. Le web est caractérisé par une forte hétérogénéité (plus de cent langues, multiples médias). La **disparité** traduit l'occurrence disséminée de l'information dans de larges collections interconnectées, engendrant des phénomènes de désorientation et de surcharge informationnelle.

Ces limitations ont catalysé plusieurs axes de recherche : l'intégration de modèles sémantiques et de bases de connaissances, le développement de systèmes de RI personnalisés et contextuels, et l'utilisation de modèles neuronaux. Ces évolutions constituent les fondements de la **recherche d'information personnalisée**, qui vise à dépasser les contraintes des approches classiques en intégrant la connaissance de l'utilisateur et de son contexte [12].

3 la Recherche d'Information Personnalisée.

Le développement du web a facilité la création, le partage et le stockage massif d'informations. Cependant, cette abondance rend l'accès à des informations pertinentes plus difficile en raison du volume, de la diversité des contenus et de l'ambiguïté des besoins des utilisateurs. Les systèmes classiques de recherche, basés sur le principe “one size fits all”, ne répondent plus efficacement aux attentes des utilisateurs. Ainsi, il devient essentiel de prendre en compte le profil, les intérêts et le contexte de l'utilisateur pour améliorer la pertinence des résultats. La recherche d'information centrée utilisateur englobe à la fois la recherche contextuelle et personnalisée, ces deux notions étant étroitement liées. Dans ce cadre, la personnalisation est considérée comme une extension de la recherche contextuelle intégrant le profil cognitif de l'utilisateur. Le chapitre traite donc de

l'émergence de la RI personnalisée, de la modélisation du profil utilisateur et de son rôle dans l'amélioration des systèmes de recherche[20].

3.1 modélisation de l'utilisateur

La **modélisation de l'utilisateur** est une discipline de recherche datant des années 1970, évoquant en premier lieu les travaux d'Allen, Cohen et Perrault [1979], dans le but d'améliorer la qualité des interactions homme-machine. Les systèmes qui adaptent leur comportement aux besoins de l'utilisateur individuel ont souvent une structure explicite de représentation qui contient des informations sur leurs utilisateurs ; cette structure s'appelle généralement un **modèle d'utilisateur**[21, 10] .

Le **modèle utilisateur** est une source de connaissances, une base de données sur un utilisateur. Plus précisément, il représente un ensemble de données persistantes qui caractérisent un utilisateur ou un groupe d'utilisateurs particuliers en contenant des caractéristiques sur les préférences, les connaissances, les objectifs, les centres d'intérêt, etc. En général, dans la littérature, les termes "modèle utilisateur" ou "profil utilisateur" signifient la même appellation.

Mais d'après Koch [2000], une différence existe :

il définit le **profil utilisateur** comme une version simple du modèle utilisateur et une collection d'informations personnelles caractérisant un utilisateur (buts, besoins, compétences, préférences, contexte d'utilisation) stockées sans interprétation, représentées par des couples (attribut, valeur).

Le **modèle utilisateur** est alors vu comme une représentation des croyances du système sur un utilisateur donné, construit à partir du profil.

Koch [2000] définit l'application du modèle utilisateur ainsi : *"Users are different : they have different background, different knowledge about a subject, different preferences, goals and interests. To individualize, personalize or customize actions a user model is needed that allows for selection of individualized responses to the user."*

Modèle utilisateur et profil utilisateur sont donc tous deux nécessaires dans le processus de personnalisation.

3.2 Pertinence utilisateur et pertinence contextuelle

La pertinence est incontestablement la question fondamentale posée lors de l'accès à l'information. Elle est une notion **subjective**, non généralisable à tout type d'information ni valide pour tout type d'utilisateurs. Elle dépend notamment du centre d'intérêt ou domaine d'application, du moment, du lieu, du support que l'utilisateur a choisi pour accéder à l'information, et du système qui délivre cette information [13].

Cette notion subjective, dépendant essentiellement du point de vue de l'utilisateur, a été l'objet d'investigations dans le cadre de la RI personnalisée. Dans ce contexte, la pertinence est spécifiée comme étant un concept **multidimensionnel**[13] dont on distingue principalement quatre types :

- **Pertinence algorithmique** : Traduite par une mesure algorithmique dépendant des caractéristiques des requêtes et des documents. C'est le seul type de pertinence indépendant du contexte.

- **Pertinence thématique** : Traduit le degré d'adéquation de l'information à couvrir le thème évoqué par le sujet de la requête. C'est le type de pertinence adressé par les assesseurs de la campagne d'évaluation TREC.

- **Pertinence cognitive** : Pertinence liée au thème de la requête, "pondérée" par la perception ou les connaissances de l'utilisateur sur ce même thème.

- **Pertinence situationnelle** : Pertinence liée à la tâche de recherche, traduisant essentiellement l'utilité de l'information relativement au but de recherche de l'utilisateur.

La **RI personnalisée** explore essentiellement la pertinence cognitive et la pertinence situationnelle. La **pertinence contextuelle** étend encore cette notion en intégrant les dimensions situationnelles dynamiques : où se trouve l'utilisateur, à quel moment cherche-t-il, avec quel dispositif, dans quel cadre social, et pour quelle tâche immédiate [22, 23].

3.3 Architecture fonctionnelle d'un SRIP

Un Système de Recherche d'Information Personnalisé (SRIP) se distingue des systèmes classiques par le fait qu'il place l'utilisateur au cœur de tout le processus d'accès à l'information. Au lieu de se contenter de modéliser les caractéristiques de l'utilisateur sous forme d'un simple profil statique, le SRIP doit être capable de déduire, à partir de ce profil, l'intention réelle de l'utilisateur au moment de sa recherche, c'est-à-dire son contexte de recherche, tout en détectant et en intégrant dynamiquement les évolutions de ses centres d'intérêt au fil du temps.

Pour atteindre cet objectif, un SRIP doit intégrer plusieurs composantes essentielles :

— Des techniques et algorithmes permettant de capturer, modéliser et représenter les buts, les préférences et les centres d'intérêt de l'utilisateur (ou d'un groupe d'utilisateurs). C'est à partir de ces éléments qu'est construit et instancié un modèle de profil utilisateur.

— Une procédure de mise à jour dynamique du profil, capable de traduire son évolution temporelle (intérêts à court terme, à long terme, détection de changements d'intérêt ou « drift »).

— Des mécanismes d'intégration du profil utilisateur dans la chaîne d'accès à l'information, permettant de reformuler la requête, de réordonner les résultats ou d'adapter la présentation afin de restituer une information réellement pertinente par rapport au profil et au contexte de l'utilisateur.

L'architecture fonctionnelle typique d'un tel système, telle que décrite par [1], est illustrée dans la figure suivante :

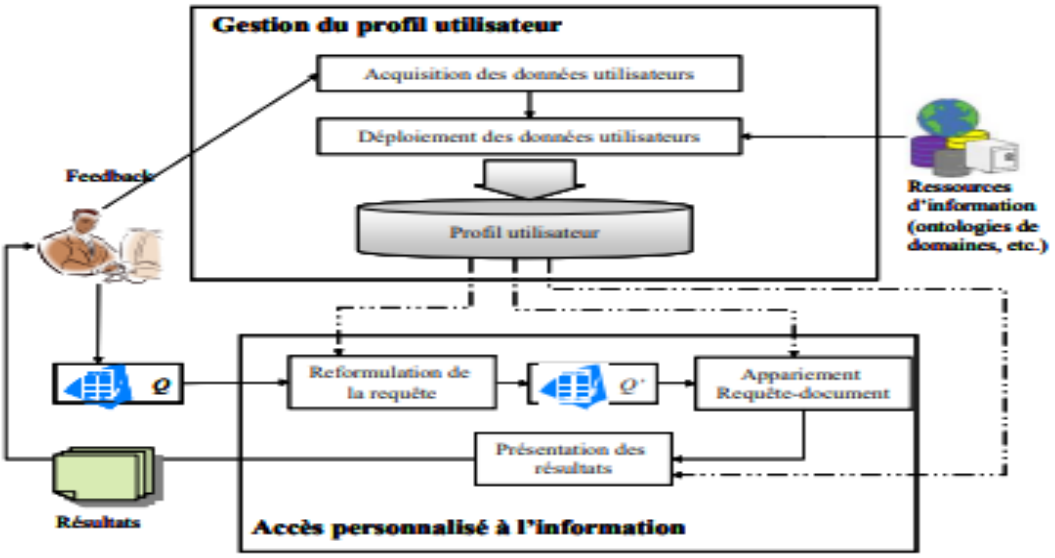


FIGURE I.1 – Architecture fonctionnelle d'un Système de Recherche d'Information Personnalisé (SRIP) d'après [1]

3.4 Profil utilisateur

Dans le but de personnaliser l'information, le SRIP (Système de Recherche d'Information Personnalisé) doit disposer des éléments ayant une incidence concrète sur la recherche en cours. Ces éléments représentent les données contenues dans le **profil utilisateur** et les métadonnées des documents. En effet, l'utilisateur et le document ne sont pas assimilés seulement à des descripteurs exprimés à l'aide de mots comme c'est le cas dans les SRI classiques ; ils possèdent tous deux des caractéristiques propres [11, 24].

Le profil utilisateur est toute structure permettant de modéliser et stocker les données caractérisant l'utilisateur au sein d'un système. Grâce à ce profil, le système possède une connaissance sur les centres d'intérêts, les préférences, et les besoins en informations de l'utilisateur, grâce auxquelles il peut lui proposer des réponses plus adaptées. Pour modéliser le profil, il faut décrire [2] :

- **Le "Quoi"** : Déterminer ce que doit représenter le contenu du profil — les informations démographiques (âge, genre, langue, profession), les centres d'intérêt thématiques, les préférences (types de documents, sources favorites), les connaissances et compétences (niveau d'expertise par domaine), et l'historique comportemental (requêtes passées, documents consultés).

- **Le "Comment"** : Déterminer la structure de contenu du profil et les techniques utilisées pour la construction de ce profil, c'est-à-dire comment représenter ces informations dans une structure exploitable par le système.

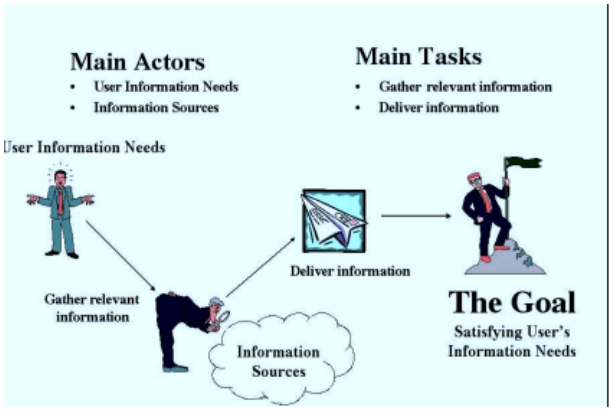


FIGURE I.2 – Le scénario de recherche d'informations [2]

3.5 Sources des données de profil

Les sources d'information pour caractériser l'utilisateur dans un système de recherche d'information contextuelle sur mobile peuvent varier : en fonction de l'aspect du contexte pris en compte (appareil mobile, profil de l'utilisateur, contexte spatial, contexte temporel, contexte de la tâche ou contexte social). Parmi les sources de données, on retrouve les spécifications techniques des équipements, des informations explicites ou sous-entendues sur les intérêts et actions de l'utilisateur, des données géographiques (GPS, Wi-Fi, Bluetooth, infrarouge...), l'heure système, les journaux des requêtes ainsi que des renseignements sociaux (contacts, amis intimes, agenda...). Les spécifications du terminal mobile recueillies grâce à des normes comme WURFL, UAProf, CC/PP et DDR (dimensions d'affichage, capacités, en-têtes HTTP...).[25]

- L'interprétation du comportement de l'utilisateur à travers des critères d'évaluation implicites comme les historiques de clics, les pages web consultées, les historiques de requêtes, les applications utilisées, les courriels et le calendrier électronique, entre autres.
- Les pages et les sites préférés, ainsi que les évaluations claires fournies par l'utilisateur (à travers des questionnaires ou des jugements directs).
- Des renseignements locaux et contextuels tels que les sources consultées, à l'instar des journaux, blogs, sites de commerce en ligne, ou encore des données de localisation et des proches identifiés par le biais de capteurs.
- Les premiers extraits ou synthèses de pages web fournis par un moteur de recherche, parfois assortis d'informations temporelles et sociales.

Les travaux combinent souvent l'acquisition explicite (précise mais contraignante) et implicite (automatique mais nécessitant plus de données) pour construire un profil utilisateur riche et adaptatif.

3.6 Acquisition des données

L'acquisition des sources d'information et des données utilisateur est l'une des problématiques importantes pour développer un SRIP efficace. On distingue deux modes d'acquisition des données utilisées dans la construction du profil utilisateur : acquisition **explicite** et acquisition **implicite** [26, 27].

3.6.1 Acquisition explicite

L'**acquisition explicite** des données utilisateur repose principalement sur les techniques de feedback explicite largement utilisées dans les systèmes de filtrage et de reformulation de requêtes par réinjection de pertinence. Ces techniques sont utilisées dans le cadre de la RI personnalisée dans le but de maintenir et faire évoluer le profil de l'utilisateur. L'acquisition explicite permet à l'utilisateur de **saisir manuellement** ses domaines d'intérêts ou de **juger explicitement** les documents renvoyés pour une requête servant à la construction de son profil [24].

Ce type d'acquisition est utilisé dans plusieurs systèmes de RI personnalisée en ligne. **Google Alertes** permet à l'utilisateur de saisir une requête-profil servant comme critère de sélection de l'information. **Google Personalized Search** (version 1.1, 2004) permet à l'utilisateur de saisir explicitement un ensemble de catégories/concepts représentant ses domaines d'intérêts. La **recherche contextuelle Yahoo ! Q** [28] permet à l'utilisateur de sélectionner explicitement un texte d'une page en cours de lecture servant à représenter un profil textuel lié à la recherche courante. L'approche de [29] requiert que l'utilisateur sélectionne parmi les concepts de l'ontologie ODP une paire de concepts (adéquat et inadéquat), utilisés dans un processus de reformulation personnalisée de la requête.

Plusieurs autres systèmes adoptent ce mode d'acquisition. [30] propose à l'utilisateur de sélectionner les concepts du profil issus de l'ODP pour la clarification du besoin. [31] et les systèmes ifWeb, Wifs et InfoWeb utilisent un feedback explicite permettant d'identifier les documents pertinents pour dériver le réseau de termes représentant le profil.

3.6.2 Acquisition implicite

L'**acquisition implicite** des données utilisateurs consiste à collecter les données de l'utilisateur en **observant ses comportements et interactions** avec le système durant les activités de recherche. L'avantage de cette approche est qu'elle ne nécessite **aucune implication directe** de l'utilisateur .

Collecte implicite des données utilisateurs : Le processus clé dans la plupart des approches de personnalisation consiste à exploiter des sources d'évidence issues de l'*historique de recherche* de l'utilisateur. Letizia, WebMate, PersonalWebWatche et OBIWAN construisent le profil en analysant les pages web visitées. D'autres sources sont également exploitées : les annotations dans Basar, les requêtes et résultats dans Syskill and Webert , et Persona, les requêtes et résumés textuels . La combinaison de sources multiples (pages web, emails, documents textes) est adoptée dans plusieurs travaux.

Indicateurs de pertinence implicite : Selon [5], toute interaction de l'utilisateur avec le système est considérée comme une estimation de son jugement d'intérêts. Ces comportements observables sont classifiés selon deux axes :la **catégorie du comportement** et l'**unité** sur laquelle se porte l'observation ,comme présenté dans le Tableau.

Behavior Category	Segment	Object	Class
Examine	View, Listen	Select	—
Retain	Print	Bookmark, Save, Delete, Purchase	Subscribe
Reference	Copy-and-paste, Quote	Forward, Reply,Link, Cite	—
Annotate	Mark up	Rate, Publish	Organize

TABLE I.2 – Potentially observable behaviors [5]

identifient certains types d'événements (défilement de page, sélection d'article) comme indicateurs de pertinence. collecte des mesures implicites basées sur les **données de clics** pour

l'apprentissage de fonctions d'ordonnancement. L'étude de [32] montre la corrélation positive entre jugements de pertinence explicites et mesures implicites en recherche sur le web.

Avantages : Aucun effort requis de l'utilisateur, grande quantité de données naturellement collectées, reflet du comportement réel.

Limites : Ambiguïté d'interprétation (un clic ne garantit pas la pertinence), bruit (clics accidentels), questions de confidentialité, démarrage lent.

3.7 Techniques de construction des centres d'intérêts

La construction du profil utilisateur consiste à organiser et extraire les éléments constitutifs du profil à partir des données collectées lors de l'étape précédente. On distingue différentes techniques basées sur l'analyse statistique de termes, le filtrage collaboratif et les techniques d'apprentissage [11].

A. Analyse statistique de termes

Cette approche analyse le contenu d'un document et extrait les mots-clés significatifs décrivant son contenu. Trois principales techniques sont distinguées :

- **Extraction d'ensemble de termes** : Techniques d'extraction des termes à partir des données visitées, servant dans l'algorithme d'apprentissage du modèle utilisateur. Systèmes représentatifs : WebMate et Alipes.

- **Extraction de réseaux de termes** : Les termes sont extraits des documents jugés et représentés sous forme de réseau de nœuds, exploitant des relations préexistantes entre termes et concepts. Le système SiteIF [Stefani & Strappavara, 1998] utilise cette technique via le dictionnaire WordNet.

- **Extraction de concepts** : Profils représentés par une hiérarchie de concepts pondérés, utilisant une taxonomie de référence(ODP (Open Directory Project), hiérarchie Yahoo). L'approche présente deux étapes :

(1) identification des concepts et niveaux de l'ontologie à exploiter.

(2) extraction des centres d'intérêts par analogie aux concepts de l'ontologie.

B. Filtrage collaboratif

Cette approche compare les profils des utilisateurs entre eux pour trouver des similarités dans leurs comportements et préférences, puis effectue des recommandations sur la base de ces similarités. Le filtrage collaboratif est caractérisé par sa flexibilité lors du passage à l'échelle et l'indépendance du format des objets manipulés. Il est utilisé dans les systèmes de recommandation où les profils sont composés de couples (objet, pertinence de l'objet) [33, 34].

C. Techniques d'apprentissage

Le principe de base est l'étude du comportement de l'utilisateur et la classification de ses caractéristiques ou celles des objets recherchés. Ces techniques sont caractérisées par leur fraîcheur et l'exactitude des données dérivées, mais présentent l'inconvénient de la complexité temporelle des algorithmes. Des exemples sont les **réseaux de neurones**, les **méthodes de classification** (raisonnement par cas, classificateurs bayésiens), et les **règles d'association** [35].

3.8 Représentation du profil utilisateur

La représentation de l'utilisateur à travers un profil permet de cibler ses besoins spécifiques et de prendre en considération ses préférences. Un modèle de représentation organise ces éléments pour faciliter leur exploitation. On distingue cinq principales approches de représentation [11, 24].

3.8.1 Représentation basée sur l'historique

Cette représentation est la plus répandue dans le domaine. L'**historique de recherche** consiste en un ensemble de requêtes, de pages Web visitées/cliquées, ou de résumés textuels des résultats accumulés au cours des sessions de recherche.

[36] représentent le profil utilisateur par son historique de recherche, servant comme base de données des requêtes soumises précédemment ainsi que leurs résultats, utilisée pour sélectionner les requêtes les plus similaires à une requête en cours.

3.8.2 Représentation ensembliste

La représentation ensembliste est basée sur un **ensemble de mots-clés pondérés** (ou vecteurs de mots-clés) représentés selon le modèle vectoriel . C'est le premier type de représentation conçu pour modéliser le profil utilisateur.

Les paquets de termes traduisent les centres d'intérêts de l'utilisateur. Le poids d'un terme est calculé selon le schéma $TF \times IDF$ et représente le degré d'intérêt de l'utilisateur. Dans [1], l'auteur distingue trois sous-modèles :

- Un ensemble de termes pondérés où chaque terme représente un centre d'intérêt possible
- Un vecteur de termes pondérés représentant un centre d'intérêt [37]
- Un ensemble de vecteurs de termes pondérés dont chacun représente un centre d'intérêt

La construction repose sur l'extraction de termes à partir des documents pertinents (jugés implicitement ou explicitement). Systèmes adoptant ce type : portails MyYahoo, InfoQuest, Anatagonomy .

3.8.3 Représentation connexionniste

La représentation connexionniste consiste à représenter les centres d'intérêts par un **réseau de nœuds pondérés**, chaque nœud représentant un concept traduisant un centre d'intérêt. Cette représentation permet de résoudre les failles de la représentation ensembliste par la mise en place de **relations de corrélation sémantiques** entre les centres d'intérêts [1].

La richesse sémantique permet de résoudre le problème de polysémie des termes, l'incohérence possible entre centres d'intérêts, et l'identification d'un profil adéquat via les relations sémantiques. [38] représentent le profil par un réseau de nœuds pondérés où chaque nœud est un concept. [39] suggère de créer les arcs reliant deux nœuds sur la base des **co-occurrences** entre ses termes.

3.8.4 Représentation multidimensionnelle

[40, 2] ont adopté ce type de représentation. Il consiste en un **ensemble de dimensions**, chacune représentant un aspect particulier (données personnelles, domaine d'intérêt, etc.). Son contenu est représenté par un modèle structuré de dimensions (ou catégories) prédéfinies ou par une structure générique où chaque dimension est constituée d'un ensemble d'attributs organisés en sous-dimensions.

Les attributs peuvent être simples ou composés. Une sous-dimension regroupe un ensemble d'attributs simples liés sémantiquement (par exemple, l'adresse est composée du numéro de rue, du nom de rue, du code postal, etc.).

3.8.5 Représentation sémantique (conceptuelle)

La **représentation conceptuelle** met en évidence une ou plusieurs relations sémantiques entre les informations du profil. Suivant une direction proposée dans , cette représentation offre

une alternative aux réseaux sémantiques. Les travaux actuels tendent à représenter le profil sous forme d'une **ontologie de concepts personnels** en se basant sur les connaissances contenues dans les ontologies plutôt que de construire les profils seulement à partir des données implicites [Chaffee & Gauch].

La représentation est essentiellement basée sur l'utilisation d'**ontologies** ou de **réseaux probabilistes**. Dans ce type d'approche, les liens entre les concepts sont explicitement induits de l'ontologie et le profil résultant inclura des relations informationnelles plus diverses et spécifiques. Contrairement à l'approche connexionniste où les centres d'intérêts sont représentés par des mots spécifiques, dans l'approche conceptuelle les nœuds correspondent à des **domaines abstraits** représentant les centres d'intérêt.

De l'association des centres d'intérêts aux concepts des domaines de l'ontologie, on obtient un profil représenté sous forme d'une **hiérarchie de concepts**. Les données implicites sont classifiées dans ces concepts, et l'intérêt de l'utilisateur pour chaque domaine est enregistré via un poids numérique [11].

Parmi les approches récentes : [41] présentent une personnalisation sémantique basée sur l'ontologie ODP. [42] proposent de modéliser le profil et les documents sous forme d'un graphe de concepts ODP avec un modèle de ré-appariement basé sur une mesure de distance sémantique.

3.9 Évolution du profil utilisateur

La **gestion de l'évolution du profil utilisateur** est un processus complémentaire à la construction du profil et désigne son adaptation à la variation des centres d'intérêt des utilisateurs au cours du temps. L'évolution se fait souvent selon un **processus incrémental** basé sur l'addition de nouvelles informations dans la représentation du profil [43, 44].

La gestion de l'évolution consiste principalement à :

— (1) **capturer les changements** des centres d'intérêts dans une première phase

— (2) **propager ces changements** au niveau de la représentation du profil

l. Cela implique soit une mise à jour de la structure/contenu des centres d'intérêts préalablement appris (changement des degrés d'intérêts), soit l'ajout d'un nouveau centre d'intérêt au profil (nouveau besoin en information) [1].

Défis de l'évolution : dérive des intérêts à long terme, intérêts temporaires (éphémères), obsolescence des anciennes données, équilibre entre stabilité et adaptabilité.

Stratégies d'évolution :

A. Fenêtre glissante temporelle : Ne conserver que les N interactions les plus récentes ou les interactions des derniers T jours [44] :

$$Profile_t = f(Interactions[t - T, t])$$

B. Décroissance temporelle : Appliquer une fonction de décroissance aux poids en fonction de leur ancienneté [45] :

$$w_i(t) = w_i(t_0) \times e^{-\lambda(t-t_0)}$$

où λ est le taux de décroissance.

C. Apprentissage incrémental : Mise à jour continue du profil à chaque nouvelle interaction, avec pondération des nouvelles données plus importante que les anciennes.

3.10 Dimensions du contexte de recherche

De nombreux travaux en recherche d'information[13, 8] ont prouvé que le concept de pertinence a un caractère **multidimensionnel** et ont identifié plusieurs critères, au-delà de la dimension thématique, pouvant affecter la perception de pertinence de l'utilisateur. Ce qui a poussé vers une direction de recherche consistant à considérer tous les critères ayant un impact sur la pertinence pour améliorer la précision des systèmes de recherche.

Pour améliorer la représentativité des profils, les approches de RI personnalisée exploitent en premier lieu les **données thématiques** issues de l'historique de recherche pertinent de l'utilisateur, qui est la dimension principale de pertinence dont tous les autres critères sont dépendants [46]. En second lieu, d'autres **capteurs supplémentaires** de pertinence liés aux différentes dimensions contextuelles sont introduits : dimension sociale, localisable, temporelle, et sémantique.

Il est à noter, comme affirmé dans [25], qu'il est difficile de trouver un système de recherche personnalisé qui gère toutes les dimensions contextuelles simultanément.

3.10.1 Dimension temporelle

La **dimension temporelle** est très importante pour la description des intérêts de l'utilisateur. La recherche d'information temporelle vise l'amélioration des modèles classiques en exploitant les informations temporelles existant dans les requêtes et les documents. Les modèles de RI combinent la notion de pertinence traditionnelle avec la dimension temporelle. En effet, dans certains cas le besoin réel d'information d'une requête est incité par un événement important occupant l'opinion publique sur les réseaux sociaux [46].

L'information temporelle peut concerner le **temps de soumission des requêtes** utilisateurs ou le **temps de publication des documents** ciblés, comme elle peut concerner les **expressions temporelles** ou celles relatives aux évènements dans les contenus des documents.

Parmi les travaux représentatifs : [47] proposent une approche combinant les informations temporelles sur le document et celles de son contenu. [48] proposent un système temporel-sémantique pour l'apprentissage de l'extraction d'événements et d'entités nommées considèrent la requête comme porteuse de deux sens : thématique et temporel. exploitent l'importance du temps et de l'expérience dans la personnalisation d'un portail pour un utilisateur mobile, proposant un profil dynamique où les centres d'intérêt sont pondérés selon des zones temporelles apprises par l'étude de la routine journalière.

3.10.2 Dimension spatiale / localisable

L'introduction de la **dimension localisation** est motivée par la prolifération de la technologie mobile (PDAs, téléphones mobiles, tablettes) qui a rendu l'accès au web possible à toute place et à tout moment. L'hypothèse de base est que la **pertinence d'un document dépend de l'emplacement de l'utilisateur** au moment de la formulation de la requête. Des études [25] montrent que **25 % des besoins informationnels** des utilisateurs mobiles sont liés à la localisation.

Pour illustrer : si un utilisateur lance la requête "Echourouk" depuis l'Algérie, il cherche probablement le journal algérien Echourouk ; s'il est en Égypte, la réponse pertinente serait le journal Echourouk d'Égypte. De même, un utilisateur en déplacement qui émet la requête "restaurant pas cher" s'intéresse davantage aux restaurants proches de sa localisation.

Parmi les travaux : [49]proposent une recommandation personnalisée dans un environnement mobile guidée par un profil situationnel basé sur la localisation.[25] propose une personnalisation basée sur un profil situationnel composé des différentes situations de recherche et les centres

d'intérêts attachés. [50]proposent un profil d'utilisateur mobile où les situations sont représentées par les lieux les plus visités, modélisées à base d'arbre. [51] traitent l'ambiguïté des requêtes par une méthode d'expansion basée sur la localisation : récupération des coordonnées spatiales, transformation en mots contextuels via un système d'information géographique, puis expansion de la requête avec les mots contextuels les plus pondérés.

3.10.3 Dimension sociale

Dans le cadre de la RI personnalisée, les **informations sociales** sont utilisées pour construire les profils des utilisateurs. Deux informations sociales principales sont exploitées.

A. Les annotations sociales (Tags)

Les tags employés par les utilisateurs pour annoter les documents sont souvent utilisés pour déterminer leurs centres d'intérêts. Ils expriment un avis, un jugement ou une impression sur les contenus des documents [52, 53]. Certaines approches proposent une représentation vectorielle des tags, dont le poids peut suivre un schéma basé sur la fréquence d'utilisation (TF) [54], un TF-IDF [55], ou une combinaison TF-IUF et BM25[56] .

Selon[57] , 50% des tags utilisés pour annoter les documents sont des termes du document.[58] ont montré que les tags sont corrélés avec les motivations d'annotation (retrouver ultérieurement, partager, etc.).

B. Les relations sociales entre utilisateurs

L'exploitation des relations sociales permet d'enrichir le profil, où la pertinence prédite d'un document peut être améliorée en explorant les actions du **voisinage** de l'utilisateur. La notion d'**importance sociale** suppose qu'un document pertinent est produit par un auteur important [59]. Des études [60] affirment que 66% de nos recherches sont similaires à celles de nos collègues.

Parmi les travaux récents : proposent dans un cadre de recherche sociale en langue arabe sur Facebook une approche sélectionnant les meilleures traces des utilisateurs (polarité, partage, j'aime) pour estimer leur importance sociale.[59] proposent une personnalisation combinant annotations sociales et termes pertinents du profil. [61] présentent une approche combinant annotations et relations sociales pour la construction du profil.[53] proposent une recherche de conversations dans les tweets basée sur la pertinence textuelle et sociale, considérant hashtags, URLs, expertise du blogueur, et influence sociale (tweets favorisés, retweetés).

3.10.4 Dimension sémantique et cognitive

Plusieurs travaux représentent le profil utilisateur à l'aide de termes ou mots-clés issus de l'historique [Uddin et al., 2012]. Néanmoins, un processus basé uniquement sur la **syntaxe** (mots-clés/termes) semble plus proche d'une opération de reconnaissance de formes que d'une opération d'assimilation et de satisfaction d'un besoin cognitif [62]. Sans l'exploitation des **liens et relations sémantiques** entre termes, la RI reste incapable de bénéficier de la richesse de la langue humaine pour une compréhension profonde de la requête [63]. Beaucoup de documents sémantiquement similaires au sujet de la requête peuvent ne pas être sélectionnés parce qu'ils utilisent d'autres termes pour décrire le même contenu.

La **sémantique** tient compte de la signification véhiculée par les termes, s'appuyant sur des ressources sémantiques externes (ontologies, thésaurus, dictionnaires) pour extraire les liens sémantiques [64].

Parmi les travaux : [65] proposent un profil spatial sémantique dans un réseau social basé sur les localisations, combinant informations de localisation et annotations sociales, utilisant "Foursquare" pour les informations sémantiques des localisations et WordNet pour filtrer les annotations.[41] présentent une personnalisation sémantique basée sur l'ontologie ODP pour

l'annotation des documents pertinents et l'indexation sémantique des documents. [42] proposent de modéliser le profil et les documents sous forme d'un graphe de concepts ODP, avec un modèle de ré-appariement basé sur une mesure de distance sémantique combinant maximum et minimum sous-graphe de concept commun. [66] intègrent dimension sociale et sémantique, sélectionnant des termes sur la base d'une similarité sémantique avec ceux de la requête.

3.11 Intégration du profil utilisateur et du contexte

À partir des informations contenues dans le profil, le SRIP va cibler un ensemble de documents. Ces informations sont représentées par une liste de termes comparée aux descripteurs des documents de la collection. Les documents ayant un degré de similarité élevé seront sélectionnés et formeront le **sous-espace de recherche** sur lequel portera la requête [67].

3.11.1 Reformulation personnalisée de la requête

La **requête** est l'expression d'un besoin circonstancié que l'utilisateur souhaite voir satisfait en tenant compte de son profil. Elle représente le **besoin à court terme** de l'utilisateur, alors que le profil représente son **besoin à long terme**.

La requête initiale contient des valeurs de paramètres du profil utilisateur fournies directement par l'utilisateur. Ces paramètres représentent les classes d'intérêt de haut niveau dans la structure hiérarchique du profil. En choisissant une classe d'intérêts, l'utilisateur exprime son but de recherche. À partir de ces choix, le système va extraire les documents correspondants et effectuer la **reformulation personnalisée de la requête**. En effet, le SRIP va considérer ces documents comme jugés pertinents afin de reformuler la requête initiale [68].

Les principales techniques de reformulation personnalisée incluent l'**expansion personnalisée de requête** (ajout de termes issus du profil de l'utilisateur aux termes de la requête originale), la **reformulation basée sur l'historique** (utilisation des requêtes passées et des documents jugés pertinents pour enrichir la requête courante), et la **reformulation basée sur les ontologies** (expansion via des concepts sémantiquement liés dans une ontologie comme ODP) [29, 30].

3.11.2 Reclassement des résultats

Dans cette phase, l'interaction entre le profil et le module de représentation du résultat se fait à travers les catégories "Customisation" et "Catégories des données de l'environnement". En fonction des informations fournies par ces catégories, le SRIP va déployer des **mécanismes d'adaptation** afin de restituer les résultats selon le contexte et les préférences de l'utilisateur.

Le **reclassement personnalisé** (personalized re-ranking) prend les résultats initiaux d'un moteur de recherche classique et les réordonne en fonction du profil utilisateur. Le score de pertinence personnalisé peut être formulé comme une combinaison linéaire du score algorithmique classique et du score de pertinence utilisateur :

$$Score_{personnalis}(d, q, u) = \alpha \cdot Score_{algo}(d, q) + (1 - \alpha) \cdot Score_{profil}(d, u)$$

où $\alpha \in [0, 1]$ contrôle l'importance relative des deux composantes, $Score_{algo}(d, q)$ est le score algorithmique classique (BM25, TF-IDF), et $Score_{profil}(d, u)$ mesure la similarité entre le document d et le profil de l'utilisateur u [69, 70].

Parmi les stratégies de reclassement : **reclassement basé sur le profil vectoriel** (calcul de la similarité cosinus entre le vecteur profil et les vecteurs documents), **reclassement basé sur les ontologies** (utilisation des relations sémantiques de l'ontologie pour mesurer la pertinence),

reclassement social (intégration des signaux sociaux comme les partages, likes, retweets), et **reclassement contextuel** (pondération des résultats selon les dimensions contextuelles : localisation, temporalité, dispositif) [59].

3.12 Évaluation des systèmes de recherche d'information personnalisée

Les méthodes d'évaluation largement adoptées en RI sont souvent basées sur une évaluation plus **quantitative que qualitative**, reposant sur la comparaison de métriques en termes de rappel et précision. Ce type d'évaluation est orienté vers une approche comparative de plusieurs systèmes reposant sur le principe d'évaluation des collections de tests. Bien qu'adopté par des campagnes d'évaluation de référence (TREC), il est contesté , notamment en raison de la non-considération du contexte de la recherche ni de la perception de la pertinence des utilisateurs. L'introduction de la dimension utilisateur accentue davantage la difficulté d'évaluation : en raison du caractère subjectif des utilisateurs et du caractère dynamique du système, il est difficile de fournir des valeurs absolues aux métriques [67, 71].

3.13 Limites de l'évaluation classique

Malgré la popularité de TREC et CLEF, ces approches présentent certaines limites[67] organisées en trois classes de problèmes :

Limites par rapport à l'utilisateur : Ces évaluations sont artificielles et arbitraires, ne tenant pas compte du contexte dans lequel se fait la recherche ni de l'utilisateur final. affirme que l'**absence de l'usager** dans le processus d'évaluation est la critique la plus importante des approches classiques.

Limites par rapport aux jugements de pertinence : La pertinence est une notion subjective. Les jugements dans TREC opèrent de façon **binaire** (pertinent/non pertinent), alors que certains documents sont plus pertinents que d'autres. montre que la pertinence considérée dans l'évaluation classique est thématique et **indépendante du contexte**, de la situation de recherche et des centres d'intérêt des utilisateurs.

Limites par rapport au corpus : Dans les corpus traditionnels, un document est évalué par rapport au nombre de documents retrouvés, alors qu'en général un utilisateur ne cherche pas des documents mais de **l'information**. constate que le protocole d'évaluation en mode batch suppose que les requêtes représentent à elles seules l'utilisateur, excluant ainsi les centres d'intérêt et interactions des utilisateurs réels.

3.14 Protocole d'évaluation

Les **protocoles d'évaluation** définissent les procédures expérimentales garantissant que les comparaisons entre systèmes sont équitables, objectives et reproductibles. Les principaux types incluent :

l'**évaluation offline** (paradigme de Cranfield, utilisant des collections statiques avec jugements préétablis, approche la plus courante en recherche académique).

l'**évaluation online** (A/B testing en condition réelle avec vrais utilisateurs, privilégiée par les entreprises comme Google et Bing), l'**évaluation centrée utilisateur** (études qualitatives avec observation directe), et les techniques de **pooling et sampling** pour les grandes collections. Le protocole consiste à comparer plusieurs systèmes en spécifiant trois composantes non indépendantes : le nombre de topics utilisés, les mesures d'évaluation utilisées, et la différence de performance requise pour considérer qu'une stratégie est meilleure qu'une autre[72, 7].

3.14.1 Protocole d'évaluation personnalisée

La problématique de l'évaluation personnalisée : Dans les SRI classiques, l'évaluation des requêtes se fait sans tenir compte du contexte de l'utilisateur. Ils supposent que les résultats retournés pour une même requête sont identiques même si exprimée par des utilisateurs différents. Considérons la requête "virus" émise par un informaticien cherchant des virus informatiques et un biologiste cherchant des virus biologiques : face à la non-considération du contexte et aux phénomènes d'accroissement du **volume** (téraoctets, pétaoctets), d'**hétérogénéité** (plus de cent langues, multiples médias) et de **disparité** (occurrence disséminée dans de larges collections), cela pose des problèmes d'ambiguïté sémantique, d'inintelligibilité des résultats et d'insatisfaction de l'utilisateur.

Ces problèmes font émerger la personnalisation comme approche essentielle, ciblant deux aspects : au **niveau de la présentation** (customisation de l'interface, couleurs, polices, positionnement) et au **niveau du contenu** (personnalisation selon les besoins et centres d'intérêts des différents utilisateurs).

Le **protocole d'évaluation personnalisée** orienté laboratoire consiste à comparer plusieurs systèmes/stratégies en spécifiant trois composantes non indépendantes [72] :

— (1) le **nombre de topics** utilisés,

— (2) les **mesures d'évaluation** utilisées,

— (3) la **différence de performance** requise pour considérer qu'une stratégie est meilleure qu'une autre.

L'évaluation de l'efficacité consiste à évaluer la liste de résultats pour chaque requête de test, à la base de la correspondance entre la pertinence algorithmique calculée par le système et la pertinence donnée par les assesseurs. L'efficacité globale est calculée comme la **moyenne des précisions** sur l'ensemble des requêtes dans la collection de test.

3.15 Mesures d'évaluation personnalisées

Différentes mesures d'évaluation ont été proposées dans le cadre des travaux sur la recherche des systèmes interactifs, pouvant également être employées pour l'évaluation d'un système d'accès personnalisé[71].

i. La mesure de précision P@X La mesure de précision P@X C'est la précision à différents niveaux de coupe. Cette précision mesure la proportion des documents pertinents retrouvés parmi les X premiers documents retournés par le système. Elle permet en particulier de s'intéresser à la haute précision, lorsque peu de documents sont restitués.

$$P@X = \frac{\text{nombre de documents pertinents dans les } X \text{ premiers résultats}}{X}$$

Avec :
X : nombre de résultats considérés (ex : 5, 10, 20. . .).
Documents pertinents : résultats corrects par rapport à la requête.

P@X mesure donc la proportion de bons résultats dans les X premiers.

ii. La mesure pertinence relative RR (Relative Relevance)

La mesure RR [Borlund & Ingwersen, 1998] a pour objectif de considérer différents types de pertinence (pertinence non binaire) dans l'évaluation de l'efficacité d'un système d'accès contextuel. Elle quantifie le **degré de concordance** entre deux ensembles de jugements (R_1 et R_2) associés à une même liste de documents. En pratique, R_1 correspond aux scores de pertinence algorithmique et R_2 à des scores de pertinence contextuelle. La corrélation est calculée via une **mesure du cosinus** :

$$RR(R_1, R_2) = \frac{\vec{R_1} \cdot \vec{R_2}}{|\vec{R_1}| \cdot |\vec{R_2}|}$$

À la différence de la précision classique, cette mesure permet de considérer les différents types de pertinence. Néanmoins, elle pose un problème lors de l'évaluation comparative car les scores algorithmiques ne sont pas étalonnés à la même échelle entre différents SRI [13].

iii. Les mesures CG (Cumulative Gain) et DCG (Discount Cumulative Gain)

Les mesures CG et DCG [18] sont des mesures **orientées position** définies dans le contexte d'une pertinence graduelle, estimant le gain de l'/utilisateur en termes de pertinence cumulée jusqu'à un rang donné :

$$CG[i] = \begin{cases} G[1] & \text{si } i = 1 \\ CG[i-1] + G[i] & \text{sinon} \end{cases} \tag{I.1}$$

$$DCG[i] = \begin{cases} G[1] & \text{si } i = 1 \\ CG[i-1] + \dfrac{G[i]}{\log_q(i)} & \text{sinon} \end{cases} \tag{I.2}$$

où $G[i]$ est la valeur de pertinence associée au document de rang i. Comparativement à CG, DCG **atténue le gain** de pertinence en fonction du rang : plus le rang est élevé, moins il est probable que l'utilisateur examine le document.

iiii. La mesure GRP (Generalised Recall and Precision)

La mesure GRP [18] généralise les mesures classiques de rappel et précision en considérant une **pertinence graduelle**. Le rappel généralisé (GR) et la précision généralisée (GP) sont calculés comme :

$$GP = \frac{\sum_{d \in R} r(d)}{|R|} \tag{I.3}$$

$$GR = \frac{\sum_{d \in R} r(d)}{\sum_{d \in D} r(d)} \tag{I.4}$$

où R est l'ensemble des documents retournés par le SRI, D est l'ensemble des documents de la collection, et $r(d)$ est la valeur de pertinence graduelle associée au document d.

De manière analogue aux mesures classiques, ces mesures peuvent être agrégées pour plusieurs requêtes ou plusieurs niveaux de rappel, donnant la possibilité de **tracer des courbes de performances**.

iv. La mesure Précision Moyenne AP (Average Precision)

La mesure AP a pour objectif de prendre en compte non seulement la présence des documents pertinents, mais aussi leur position dans le classement, ce qui la rend plus représentative de l'expérience utilisateur. Elle quantifie la **qualité du classement** en calculant la moyenne des précisions obtenues à chaque rang où un document pertinent est retrouvé.

À la différence de la précision classique, la mesure AP récompense fortement les systèmes qui placent les documents pertinents en haut de la liste des résultats. Elle est définie par la formule suivante :

$$AP = \frac{1}{N_{rel}} \sum_{k=1}^{n} P@k \times rel(k)$$

Où N_{rel} représente le nombre total de documents pertinents dans la collection, $P@k$ la précision calculée au rang k, $rel(k)$ une fonction indicatrice (qui vaut 1 si le document au rang k est pertinent, et 0 sinon), et n le nombre total de documents retournés. La moyenne des AP sur plusieurs requêtes est appelée **MAP (Mean Average Precision)** et constitue une mesure globale de référence pour comparer différents systèmes.

4 Conclusion

Ce chapitre a établi les fondements théoriques et pratiques nécessaires à la compréhension des deux paradigmes centraux de la recherche d'information : le paradigme classique et le paradigme personnalisé et contextuel.

Dans un premier temps, nous avons exploré les bases de la **RI classique** : ses notions fondamentales (terme, document, requête, collection, pertinence), son processus opérationnel (expression du besoin, indexation avec ses étapes de tokenization, élimination des mots vides, normalisation et pondération TF-IDF/BM25, reformulation, appariement), ses modèles formels (booléen, vectoriel, probabiliste avec BM25, connexionniste avec MERCURE, LSI, et neuronaux), ses méthodes d'évaluation standardisées (collections TREC, mesures de précision, rappel, F-mesure, MAP, MRR, nDCG), et enfin ses limites fondamentales liées à la dépendance lexicale, l'ambiguïté, et l'absence de personnalisation [7].

Dans un second temps, nous avons présenté les mécanismes de **modélisation de l'utilisateur** : la notion de pertinence utilisateur multidimensionnelle (algorithmique, thématique, cognitive, situationnelle), la distinction entre profil et modèle utilisateur, les sources et modes d'acquisition des données (explicite et implicite avec leurs indicateurs de pertinence), les techniques de construction des centres d'intérêts (analyse statistique de termes, filtrage collaboratif, techniques d'apprentissage), et les cinq approches de représentation du profil (historique, ensembliste, connexionniste, multidimensionnelle, sémantique/conceptuelle), ainsi que les stratégies d'évolution temporelle du profil. Nous avons également décrit les quatre dimensions contextuelles (temporelle, spatiale/localisable, sociale, sémantique), les mécanismes d'intégration du profil (reformulation personnalisée et reclassement), et les méthodes d'évaluation spécifiques aux systèmes personnalisés (RR, CG, DCG, GRP) [11, 8].

L'ensemble de ces concepts constitue le socle théorique sur lequel s'appuient les approches modernes de personnalisation, et notamment les travaux utilisant les **Large Language Models (LLMs)** pour la modélisation de l'utilisateur et l'identification du contexte de recherche — sujet central des chapitres suivants de cette mémoire.

Chapitre II :

Large Language Models (LLMs)

1 Introduction

L'avènement des Large Language Models (LLMs) constitue une rupture paradigmatique majeure dans le domaine du traitement automatique du langage naturel (TALN) et, plus largement, dans l'intelligence artificielle.Un **Large Language Model (LLM)** est un système d'intelligence artificielle basé sur des techniques d'apprentissage profond, conçu pour traiter de vastes quantités de données textuelles afin de comprendre et générer du langage naturel. Ces modèles, fondés sur l'architecture Transformer et entraînés sur des corpus de données textuelles d'une échelle sans précédent, ont démontré des capacités remarquables de compréhension, de génération, de raisonnement et d'adaptation à de nouveaux contextes souvent sans nécessiter de réentraînement spécifique à la tâche [73].

Depuis la publication de GPT-3 par OpenAI en 2020 [74], le développement des LLMs a connu une accélération spectaculaire, avec l'émergence successive de modèles toujours plus performants : PaLM [75], LLaMA [76], Mistral [77], Claude, Gemini, et bien d'autres. Ces modèles partagent des fondements architecturaux communs tout en se distinguant par leur taille, leurs données d'entraînement, leurs techniques d'alignement et leurs domaines d'application privilégiés.

Ce chapitre présente les fondements théoriques et pratiques des LLMs. Nous commençons par retracer l'évolution architecturale qui a conduit aux LLMs modernes, depuis les réseaux de neurones récurrents jusqu'à l'architecture Transformer. Nous décrivons ensuite le processus d'entraînement, les techniques d'adaptation, et les paradigmes d'utilisation émergents notamment les agents LLM et l'architecture RAG (Retrieval-Augmented Generation) , avant d'aborder spécifiquement leur application à la recherche d'information personnalisée

2 Définition des LLMs

Les LLM (Large Language Models) sont des systèmes d'intelligence artificielle construits sur des architectures de réseaux de neurones profonds, plus précisément des transformateurs, et entraînés sur des corpus textuels d'une ampleur considérable. Leur puissance repose sur des techniques avancées d'apprentissage profond (deep learning), qui leur permettent d'assimiler les subtilités et la complexité du langage naturel à partir de l'analyse statistique de vastes ensembles de données textuelles. les LLM constituent des outils d'une grande sophistication, capables de comprendre et de générer le langage humain avec un niveau de précision et de pertinence contextuelle sans précédent dans l'histoire de l'intelligence artificielle.

Exemples de LLMs populaires

- **Famille GPT :** A famille des modèles GPT développée par OpenAI a connue une évolution rapide depuis le lancement de GPT-1 en 2018. Basé sur l'architecture Transformer, en 2018 GPT-1 a introduit le concept du pré-entraînement sur de grands corpus textuels. GPT-2 a ensuite amélioré les capacités de génération de texte grâce à une augmentation importante du nombre de paramètres. En 2020, GPT-3 a marqué une avancée majeure avec 175 milliards de paramètres et des capacités d'apprentissage en contexte. À partir de GPT-3, plusieurs variantes spécialisées ont été développées, notamment Codex pour la génération de code et ChatGPT pour les interactions conversationnelles. En 2023, GPT-4 a introduit des capacités multimodales capables de traiter le texte et les images avec une meilleure précision et un meilleur alignement éthique. Enfin, les modèles GPT-5 et GPT-5.5,

apparus entre 2025 et 2026, ont apporté des améliorations importantes en raisonnement, compréhension contextuelle, multimodalité et génération de code, renforçant ainsi les performances des systèmes d'intelligence artificielle modernes.

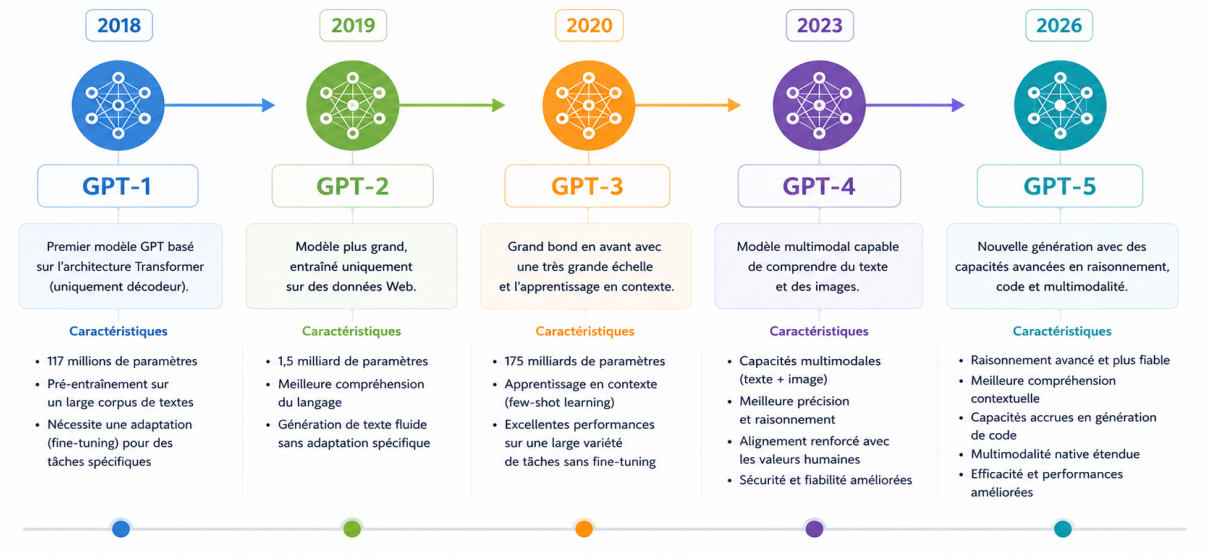


FIGURE II.1 – Évolution des modelés GPT d'OpenAI.

- **Famille LLaMA :** La famille des modèles Meta LLaMA a connu une évolution rapide entre 2023 et 2026. Lancé en 2023, LLaMA 1 a introduit une série de modèles de langage open-source performants destinés à la recherche en intelligence artificielle. La même année, LLaMA 2 a amélioré les performances, le contexte de traitement et l'accessibilité pour les applications commerciales. En 2024, LLaMA 3 a marqué une avancée importante grâce à de meilleures capacités de raisonnement, de génération de code et de compréhension multilingue. En 2025, LLaMA 4 a introduit des fonctionnalités multimodales plus avancées, capables de traiter simultanément le texte, l'image et d'autres types de données. Enfin, les modèles prévus pour 2026, souvent associés à LLaMA 5, visent à offrir une meilleure efficacité, un raisonnement plus puissant et une compréhension contextuelle étendue, renforçant ainsi le rôle des modèles open-source dans le domaine des grands modèles de langage (LLMs).

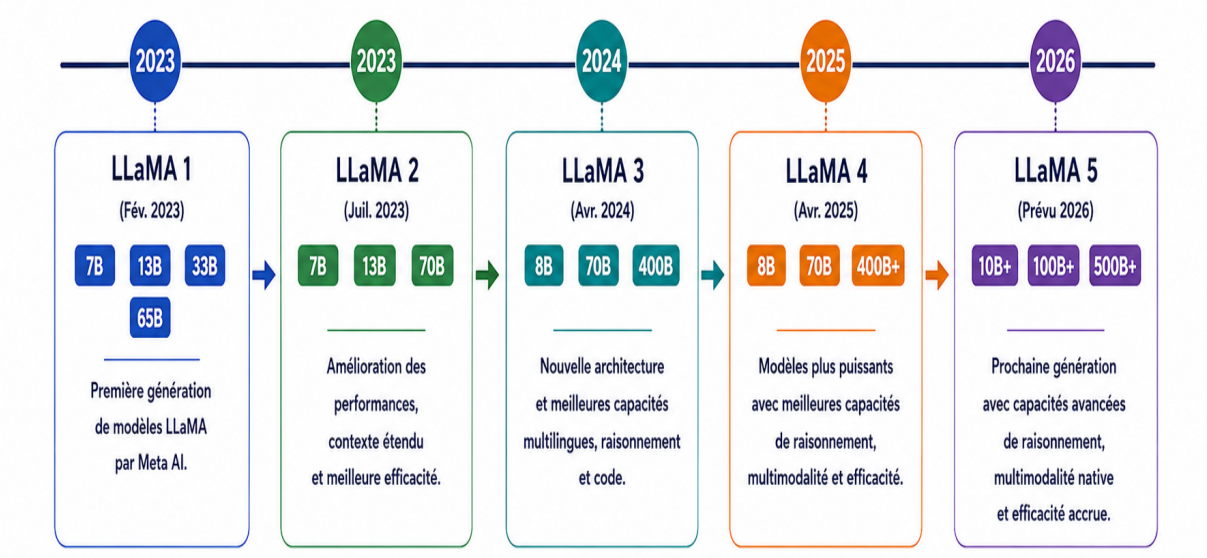


FIGURE II.2 – Famille des modèles LLaMA.

3 Évolution des modèles de langage

Les Large Language Models (LLMs) constituent une catégorie avancée de modèles d'intelligence artificielle capables de comprendre, traiter et générer du texte en langage naturel. Ces modèles sont entraînés sur d'immenses corpus de données textuelles afin d'apprendre les structures linguistiques, le contexte sémantique et les relations entre les mots. L'évolution des LLMs s'est réalisée progressivement à travers plusieurs étapes technologiques majeures, allant des approches symboliques fondées sur des règles jusqu'aux architectures neuronales modernes basées sur les Transformers.

• **1950 – 1980** : Les modèles à base de règles

Durant cette période, les systèmes de traitement du langage naturel reposaient principalement sur des règles linguistiques écrites manuellement par des experts comme le premier chatbot **ELIZA** (1966). Ces approches symboliques utilisaient des dictionnaires et des règles grammaticales fixes pour analyser ou générer du texte. Malgré quelques résultats encourageants, ces systèmes restaient limités face à la complexité et à l'ambiguïté du langage naturel[78].

• **1980 – 2000** : La renaissance des approches statistiques

Les années 1980 marquent l'introduction des méthodes statistiques dans le traitement automatique du langage naturel, notamment les modèles **Bag-of-Words** et les modèles **n-grams**. À la fin des années 1980 apparaissent les premiers réseaux de neurones récurrents **RNNs**, capables de traiter des séquences textuelles, bien que leurs performances soient limitées par la faible puissance de calcul disponible. Durant les années 1990, le développement de grands corpus annotés favorise l'émergence de tâches telles que l'étiquetage morpho-syntaxique (POS Tagging) et la reconnaissance des entités nommées (NER).

• **2000 – 2015** : L'ère du Big Data et du Deep Learning

Avec l'essor d'Internet, des réseaux sociaux et du Big Data, les modèles neuronaux connaissent une progression importante. En 2010, le modèle **RNNLM** améliore la génération de texte grâce à une meilleure prise en compte du contexte séquentiel. En 2013, l'introduction de Word2Vec révolutionne la représentation vectorielle des mots. Ensuite, en 2015, GNMT (Google Neural Machine Translation) démontre l'efficacité de la traduction neuronale à grande échelle en exploitant les techniques du Deep Learning.

• **2017 – 2026** : La révolution des Transformers et des LLMs modernes
L'année 2017 constitue un tournant majeur avec l'introduction de l'architecture **Transformer** par Google Research, permettant un traitement parallèle plus efficace des séquences textuelles. En 2018, OpenAI lance GPT-1, un modèle de 117 millions de paramètres basé sur cette architecture. En 2019, BERT développé par Google améliore considérablement la compréhension contextuelle du langage. En 2020, GPT-3 marque une avancée spectaculaire avec 175 milliards de paramètres et des capacités multitâches avancées[79].

À partir de **2022**, les modèles conversationnels deviennent largement accessibles au grand public grâce à ChatGPT. La même période voit également l'émergence de modèles performants tels que Anthropic avec **Claude**, Meta avec **LLaMA**, ainsi que Google DeepMind avec **Gemini**. En 2024 et 2025, les LLMs deviennent multimodaux, capables de traiter simultanément du texte, des images, de l'audio et de la vidéo. En 2026, ces modèles sont largement intégrés dans les assistants intelligents, les moteurs de recherche, la programmation assistée, la recherche scientifique et les systèmes décisionnels avancés, confirmant leur rôle central dans l'évolution de l'intelligence artificielle moderne[80].

4 Architecture Transformer et mécanismes fondamentaux

1.Représentation des entrées : tokenisation et embeddings
Avant d'être traitée par le Transformer, toute séquence textuelle est découpée en tokens par un algorithme de tokenisation généralement BPE (Byte Pair Encoding) [81] ou WordPiece [79]. Chaque token est ensuite converti en un vecteur dense de haute dimension (embedding) grâce à une couche d'embedding apprise lors de l'entraînement. À ce vecteur s'ajoute un encodage positionnel qui informe le modèle de la position de chaque token dans la séquence, compensant l'absence de récurrence de l'architecture.

2.Mécanisme d'attention multi-têtes
Le mécanisme d'attention est le composant central du Transformer [82]. Pour une séquence d'entrée, il calcule pour chaque token une représentation contextualisée en pondérant les contributions des autres tokens selon leur pertinence. Formellement, l'attention est définie par trois matrices de projection : Q (Query), K (Key) et V (Value), telles que :

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) \cdot V \tag{II.1}$$

où d_k est la dimension des vecteurs clés. L'architecture multi-têtes réplique ce mécanisme en parallèle avec h têtes d'attention différentes, permettant au modèle de capturer simultanément différents types de relations sémantiques et syntaxiques entre les tokens [82].

3.Structure en couches et fenêtre de contexte
Un LLM de type décodeur (GPT-like) est composé d'un empilement de N couches identiques, chacune contenant un bloc d'attention propre à la couche (Self-Attention) et un réseau de neurones entièrement connecté (Feed-Forward Network). Des couches de normalisation (Layer Norm) [83] et des connexions résiduelles assurent la stabilité de l'entraînement [82, 83].

La fenêtre de contexte (context window) désigne le nombre maximal de tokens que le modèle peut traiter simultanément. Cette limite, initialement de 2048 tokens pour GPT-3, a été considérablement étendue dans les modèles récents : 128K tokens pour GPT-4 Turbo [84], jusqu'à 1M pour certaines variantes de Gemini 1.5 [85]. Cette extension est cruciale pour la

recherche d'information, car elle permet d'intégrer des documents plus longs dans le contexte de génération.

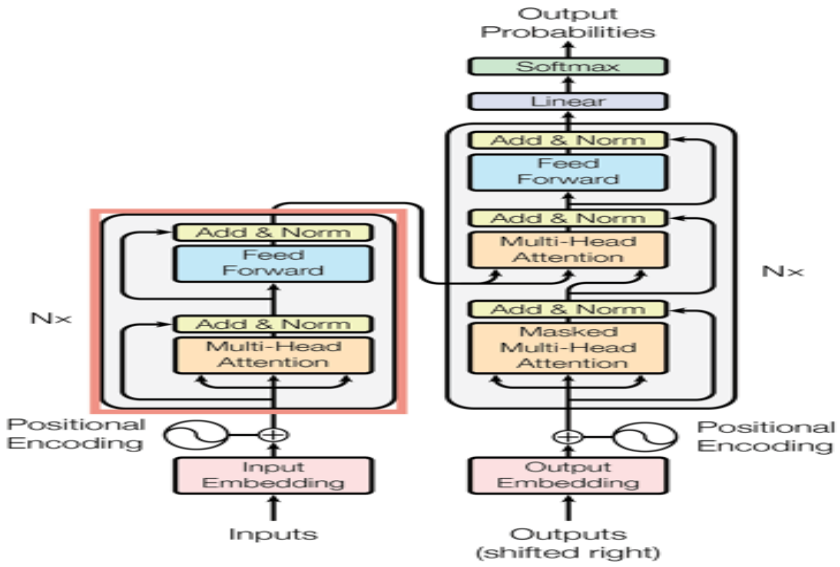


FIGURE II.3 – The Transformer - model architecture.

5 Processus d'entraînement des LLMs

1.Pré-entraînement sur corpus massifs
Le pré-entraînement consiste à optimiser le modèle sur un objectif auto-supervisé , typiquement la prédiction du prochain token (Causal Language Modeling) pour les modèles génératifs, ou la reconstruction de tokens masqués (Masked Language Modeling) pour les modèles encodeurs [79, 86]. Les corpus utilisés combinent des textes issus du Web crawlé (Common Crawl, C4), de livres (Books3), de code (GitHub), et d'encyclopédies (Wikipédia), représentant au total des centaines de milliards à plusieurs trillions de tokens [74, 87].

2.Alignement par renforcement humain (RLHF) Le pré-entraînement seul produit un modèle capable de compléter des textes, mais pas nécessairement de suivre des instructions ou de produire des réponses utiles et inoffensives. L'alignement par Reinforcement Learning from Human Feedback (RLHF) [3] résout ce problème en trois étapes :

— (1) collecte de démonstrations humaines pour un fine-tuning supervisé initial (SFT).

— (2) entraînement d'un modèle de récompense (reward model) à partir de comparaisons humaines entre réponses .

— (3) optimisation par PPO (Proximal Policy Optimization) en maximisant le score de récompense tout en pénalisant la dérive du modèle par rapport à sa version SFT.

Des variantes plus simples et plus stables ont ensuite été proposées, notamment DPO (Direct Preference Optimization) [88], qui reformule l'objectif RLHF comme un problème de classification binaire sur des paires de réponses préférées/rejetées, sans nécessiter de modèle de récompense explicite.

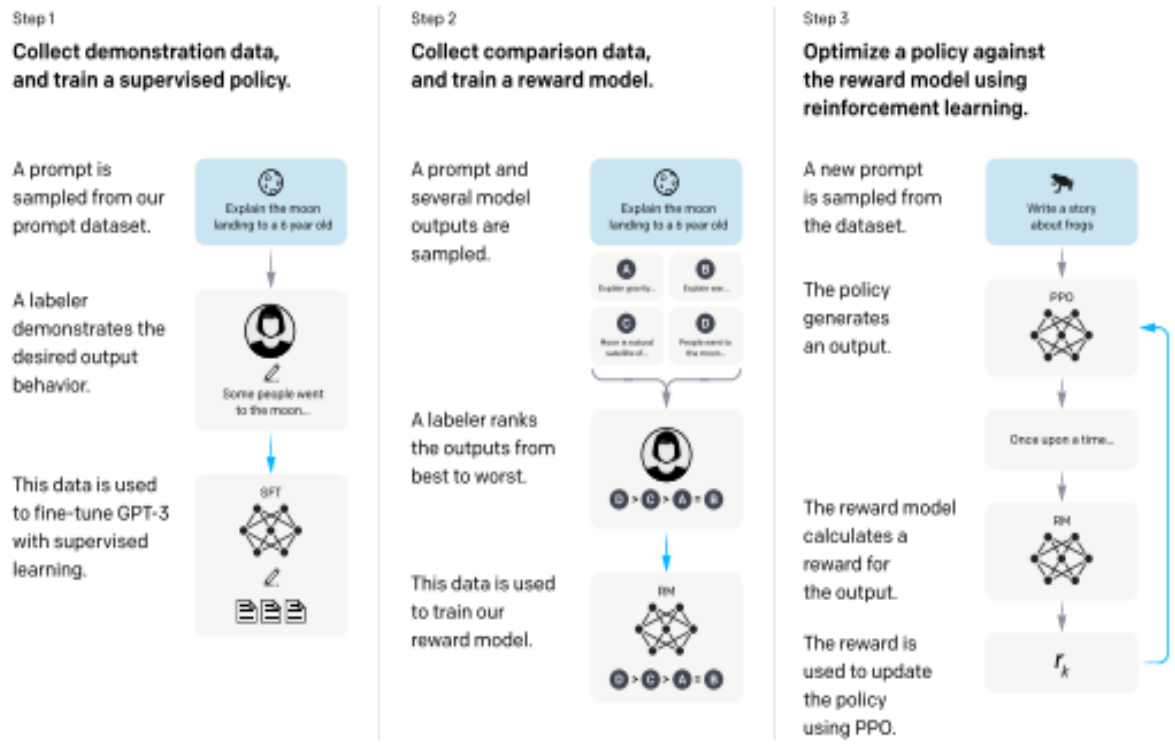


FIGURE II.4 – Un diagramme illustre les trois étapes de notre méthode : (1) ajustement fin supervisé (SFT), (2) entraînement du modèle de récompense (RM), et (3) apprentissage par renforcement via l'optimisation de politique proximale (PPO) sur ce modèle de récompense. Les flèches bleues indiquent que ces données sont utilisées pour entraîner l'un de nos modèles. À l'étape 2, les cases A à D représentent des exemples de nos modèles classés par les étiqueteurs.[3]

3.Fine-tuning et adaptation efficace (PEFT / LoRA) : Le fine-tuning complet d'un LLM est coûteux en ressources. Des techniques d'adaptation efficaces en paramètres (PEFT) ont été développées pour adapter les LLMs à des tâches ou domaines spécifiques en ne modifiant qu'une faible fraction de leurs paramètres. La technique la plus répandue est LoRA (Low-Rank Adaptation) [89], qui décompose les mises à jour de poids en produits de matrices de faible rang, réduisant le nombre de paramètres entraînables de plusieurs ordres de grandeur. Des extensions comme QLoRA [90] permettent d'entraîner des modèles quantifiés sur des GPU grand public.

6 Capacités des LLMs : in-context learning et raisonnement

1. In-context learning et prompting :

L'une des propriétés les plus remarquables des LLMs est leur capacité à apprendre à partir du contexte du prompt (in-context learning) sans mise à jour de paramètres [74]. On distingue trois régimes :

— (1) zero-shot : la tâche est décrite sans exemple .

— 2) few-shot : quelques exemples entrée/sortie sont fournis dans le prompt.

— (3) chain-of-thought (CoT) : le modèle est invité à expliciter ses étapes de raisonnement avant de donner sa réponse finale [91], améliorant significativement les performances sur des tâches de raisonnement complexe.

2. Capacités de raisonnement et hallucinations :

Les LLMs modernes manifestent des capacités de raisonnement étendues : raisonnement arithmétique, inférence logique, planification et décomposition de tâches [92]. Cependant, ils souffrent du phénomène d'hallucination , la génération de contenus factuellement incorrects présentés avec confiance [93]. Ce problème est particulièrement critique en recherche d'information, où la fiabilité des réponses est primordiale. L'architecture RAG (cf. §2.7) constitue la principale solution architecturale à ce problème, en ancrant la génération dans des documents récupérés depuis une base de connaissances externe.

7 Agents LLM

• **Définition :**

Un agent LLM (Large Language Model Agent) est un système informatique autonome ou semi-autonome qui utilise un grand modèle de langage (LLM) comme composant central, souvent qualifié de « cerveau » ou de moteur de raisonnement. Sa fonction principale est de percevoir son environnement (numérique ou potentiellement physique via des capteurs), de prendre des décisions, de planifier des séquences d'actions complexes et d'interagir avec des outils logiciels, des API ou d'autres systèmes externes pour accomplir des objectifs spécifiques définis par un utilisateur ou par sa programmation intrinsèque. Contrairement à un LLM standard qui se contente de générer du texte en réponse à une invite, un agent LLM engage une boucle d'interaction dynamique : il utilise le LLM pour raisonner sur une tâche, choisir une action ou un outil pertinent, exécuter cette action, observer le résultat, puis utiliser cette nouvelle information pour planifier l'étape suivante, jusqu'à l'atteinte de l'objectif final [94, 95].

• **Architecture générale d'un agent LLM :**

L'architecture d'un agent LLM repose sur une boucle de fonctionnement itérative composée de plusieurs modules interdépendants qui collaborent pour résoudre une tâche.

— **Perception** : l'agent reçoit l'état courant de l'environnement la requête initiale, les résultats des actions précédentes, les retours utilisateur, ou les données issues de capteurs et d'API .

— **Raisonnement et planification** : le LLM analyse cet état, décompose la tâche en sous-objectifs si nécessaire, et sélectionne l'action la plus appropriée parmi les outils disponibles .

— **Action et exécution** : l'agent déclenche l'outil choisi (moteur de recherche, interpréteur de code, base de données, API externe, navigateur web) et collecte le résultat.

— **Intégration de l'observation** : le résultat est ajouté au contexte courant, et le cycle recommence jusqu'à ce que l'objectif soit atteint ou qu'une condition d'arrêt soit satisfaite.

Cette architecture en boucle confère aux agents des capacités qualitativement supérieures à celles des LLMs non agentiques : résolution de tâches multi-étapes, adaptation dynamique aux imprévus, utilisation de sources d'information en temps réel, et maintien d'un état interne cohérent sur de longues interactions [96].

•Composants fondamentaux d'un agent LLM :

Un agent basé sur un grand modèle de langage repose sur un ensemble de composants essentiels qui lui permettent d'interagir avec son environnement, de raisonner, d'agir et de s'adapter à des tâches complexes. Ces composants peuvent être organisés en quatre blocs principaux.

a) Le LLM comme moteur de raisonnement.

Le modèle de langage est le cœur cognitif de l'agent. Il examine l'état actuel de la tâche et donne soit une réponse finale, soit une décision intermédiaire orientée vers une action. Ses performances reposent largement sur ses capacités de raisonnement, notamment grâce à des approches comme le chain-of-thought [91] qui décomposent le raisonnement en plusieurs étapes explicites.

Plus le modèle est performant, plus les chaînes de raisonnement qu'il génère sont cohérentes et fiables, ce qui améliore directement la qualité des décisions et la planification des actions dans des environnements complexes.

b) Le système de mémoire.

La mémoire constitue un élément central permettant à l'agent de conserver et d'exploiter des informations au-delà du contexte immédiat. Elle est généralement structurée en plusieurs niveaux complémentaires [97] :

— **Mémoire à court terme (in-context)** :il s'agit des informations disponibles dans la fenêtre de contexte courante du modèle, notamment l'historique de la conversation, les résultats récents des actions et les directives du système. Elle est donc temporaire et ne persiste pas au-delà de la session.

— **Mémoire à long terme externe** :elle repose sur des bases de données vectorielles (FAISS, Chroma, Weaviate. . .) . Les informations y sont stockées sous forme d'embeddings et sont récupérables par similarité sémantique, permettant ainsi une mémoire persistante et exploitable à long terme.

— **Mémoire épisodique** : est celle qui enregistre les séquences d'expériences passées (ce qui a marché, les erreurs, les stratégies mises en œuvre). Elle permet à l'agent de tirer parti des expériences passées similaires pour améliorer ses décisions futures.

— **Mémoire sémantique** : correspond aux connaissances générales intégrées dans les paramètres du modèle lors de la phase de pré-entraînement. Elle est une base de connaissances stable qui permet de comprendre et de raisonner sur des requêtes variées.

c) Les outils (tool use)
Un agent peut être équipé d'un ensemble d'outils hétérogènes : moteurs de recherche web, bases de données vectorielles, interpréteurs Python, APIs REST, navigateurs web, systèmes de fichiers, ou outils de calcul symbolique. La capacité à sélectionner et à appeler le bon outil au bon moment est l'une des compétences clés d'un agent performant [98].

d) Le contrôleur d'agent (orchestrateur)
Il coordonne le flux d'exécution : gestion des prompts, routage vers les outils, agrégation des résultats, et détection des conditions d'arrêt. Des frameworks comme LangChain [99], LlamaIndex [100] et AutoGen [101] encapsulent ces fonctions dans des abstractions réutilisables.

7.1 Paradigmes de raisonnement agentique

Le raisonnement agentique repose sur plusieurs paradigmes qui permettent aux Agents LLM de combiner réflexion, planification et interaction avec l'environnement. Les principaux sont les suivants :

• **ReAct : Reasoning and Acting**
Le paradigme ReAct [102] est l'un des premiers à avoir formalisé l'entrelacement du raisonnement et de l'action dans un flux unifié. À chaque étape, le LLM génère successivement :

— (1) une *Thought* : une trace de raisonnement explicite décrivant son interprétation de la situation et son intention

— (2) une *Action* : l'outil à appeler et les paramètres correspondants

— (3) une *Observation* :le résultat retourné par l'outil, intégré dans le contexte pour l'étape suivante.

Cette approche a démontré des performances supérieures à celles des méthodes de raisonnement seul (chain-of-thought) ou d'action seule sur des benchmarks de question-réponse multi-sources comme HotpotQA et FEVER [102]. La transparence des traces de raisonnement facilite également le débogage et l'explication des décisions.

• **Plan-and-Execute**
L'approche Plan-and-Execute [103] sépare explicitement la phase de planification de la phase d'exécution. Un premier appel au LLM produit un plan structuré listant toutes les sous-tâches à accomplir. Un second composant (agent exécuteur) traite séquentiellement chaque sous-tâche. Cette séparation améliore la gestion des tâches longues et permet une révision dynamique du plan si une sous-tâche échoue ou produit un résultat inattendu.

• **Réflexion et auto-correction**
Des travaux récents introduisent des mécanismes de réflexion permettant à l'agent d'évaluer la qualité de ses propres réponses et de se corriger. Reflexion [104] maintient une mémoire épisodique verbale des tentatives précédentes et utilise un signal de feedback pour générer des plans de correction. Self-Refine [105] fait itérativement réviser et affiner ses réponses par le modèle lui-même via une boucle de critique-révision.

7.2 Domaines d'application

Les agents basés sur les grands modèles de langage sont aujourd'hui utilisés dans plusieurs domaines applicatifs grâce à leur capacité à combiner raisonnement, compréhension du langage naturel et utilisation d'outils externes pour résoudre des tâches complexes [94, 95, 96].

Dans **la recherche scientifique**, ils sont exploités pour explorer la littérature, synthétiser des résultats provenant de différentes sources et assister la génération de nouvelles hypothèses de

recherche. Ils contribuent ainsi à accélérer le processus de découverte et à améliorer l'exploitation des connaissances existantes [106].

Dans le développement logiciel, les agents LLM permettent d'automatiser plusieurs étapes du cycle de développement, notamment la génération de code, la détection et la correction de bugs ainsi que l'exécution de tests. Ils interagissent directement avec des environnements de programmation et des dépôts de code pour assister les développeurs dans des tâches réelles [107].

Dans **le domaine médical**, ces agents jouent un rôle d'assistance au raisonnement clinique en analysant et en synthétisant des informations issues de sources médicales fiables. Ils aident à croiser les données scientifiques afin de soutenir la prise de décision diagnostique et thérapeutique [108].

Dans **le domaine financier**, ils sont utilisés pour analyser des données de marché, interpréter des rapports économiques et produire des synthèses utiles à la prise de décision. Leur capacité à traiter des données dynamiques leur permet d'être adaptés aux environnements financiers en constante évolution [109].

Enfin, dans **le domaine de la robotique et des systèmes cyber-physiques**, les agents LLM permettent de transformer des instructions en langage naturel en actions exécutables. Ils interviennent dans la planification et le contrôle de tâches physiques, facilitant l'interaction entre l'humain et les systèmes robotiques [110].

7.3 Systèmes multi-agents

Une extension naturelle des agents individuels est l'organisation de systèmes multi-agents (SMA) dans lesquels plusieurs agents LLM spécialisés collaborent, débattent ou se corrigent mutuellement [101, 111]. Cette organisation présente plusieurs avantages par rapport aux agents uniques :

- **Division du travail** : chaque agent est spécialisé dans un sous-domaine ou un type de tâche, ce qui améliore les performances globales .
- **Vérification croisée** : plusieurs agents peuvent évaluer indépendamment un résultat avant validation, réduisant les erreurs et hallucinations .
- **Parallélisme** : des sous-tâches indépendantes peuvent être traitées simultanément, réduisant la latence totale .
- **Robustesse** : la défaillance d'un agent n'interrompt pas nécessairement l'ensemble du système.

AutoGen [101] propose un framework de conversation entre agents personnalisables, permettant des patterns de collaboration flexibles ,débat, critique-révision, délégation hiérarchique. Society of Mind [111] formalise ces interactions en s'inspirant des théories cognitives pour structurer la coopération entre agents spécialisés.

8 Agents LLM dans la recherche d'information

La convergence entre agents LLM et recherche d'information représente une évolution qualitative des systèmes de RI. D'un côté, la RI dispose de méthodes éprouvées d'indexation, de représentation documentaire et d'évaluation ; de l'autre, les agents LLM apportent compréhension sémantique profonde, planification adaptative et personnalisation dynamique. La combinaison de ces deux paradigmes constitue l'architecture de référence des systèmes de RI de nouvelle génération [112, 113].

8.1 Limites de la recherche d'information classique adressées par les agents LLM

Le tableau ci-dessous met en correspondance les limitations identifiées dans la RI classique (§1.2.5) avec les solutions apportées par les agents LLM :

Limitation de la RI classique	**Mécanisme agentique compensateur**	**Référence**
Vocabulary mismatch entre requête et document	Reformulation et expansion de requête par LLM ; encodage dense (DPR)	[114, 115]
Requête traitée comme signal figé	Décomposition en sous-requêtes itératives ; requêtes de suivi adaptatives	[116]
Pertinence uniquement algorithmique/thématique	Profil utilisateur intégré au prompt ; personnalisation du reranking	[117]
Ignorance du contexte utilisateur	Mémoire long terme du profil ; dimensions temporelle, spatiale, sociale	[118]
Réponse = liste de documents	Génération d'une réponse synthétique sourcée (RAG)	[119]
Pas d'exploration proactive	Agent décide de chercher, affine, et itère selon les observations	[116]
Évaluation indépendante de l'utilisateur	Métriques personnalisées : nDCG@k personnalisé, satisfaction utilisateur	[120]

TABLE II.1 – Limites RI classique vs solutions agentiques

8.2 Architecture d'un système de RI augmenté par agents LLM

Un système de RI agentique complet est composé de plusieurs composants fonctionnels interconnectés qui collaborent de manière orchestrée pour produire des résultats personnalisés et pertinents [113, 121]. Chaque composant joue un rôle spécifique dans le processus global de recherche..

• **Composant de compréhension et reformulation de la requête :**
Lorsqu'une requête utilisateur est soumise, elle n'est pas directement transmise au moteur de recherche. Elle est d'abord traitée par un composant de compréhension sémantique.
Ce composant identifie : l'intention de recherche (informationnelle, navigationnelle, transactionnelle), détecter les ambiguïtés, et enrichir la requête avec des termes connexes issus de sa connaissance paramétrique [122].

Plusieurs techniques de reformulation ont été développées : la génération de requêtes hypothétiques (HyDE) [123] fait générer au LLM un document hypothétique répondant à la requête, puis utilise cet embedding comme vecteur de recherche résolvant ainsi le problème de vocabulary mismatch de manière élégante. La décomposition en sous-questions [124] divise les requêtes complexes ("Compare l'impact de GPT-4 et de LLaMA sur les applications médicales") en sous-requêtes atomiques traitées indépendamment puis fusionnées.

• **Composant de retrieval hybride :**
Le retrieval dans les systèmes agentiques modernes combine deux approches complémentaires [125] :

— **Retrieval sparse (BM25, TF-IDF)** : efficace pour la correspondance exacte de termes, très rapide, robuste aux entités nommées et aux identifiants techniques .

— **Retrieval dense (DPR, E5, BGE)** : capture la similarité sémantique même en l'absence de termes communs, mais plus coûteux en calcul .

— **Fusion hybride ((RRF)Reciprocal Rank Fusion)** : combine les listes de résultats sparse et dense en pondérant les rangs, offrant les avantages des deux approches [126].

L'agent peut également décider dynamiquement quelle stratégie de retrieval employer selon la nature de la requête : une question factuelle précise privilégiera le retrieval sparse, une question sémantique large privilégiera le retrieval dense.

• **Composant de reranking et filtrage :**
Les documents récupérés par le retrieval sont ensuite reclassés par un cross-encoder, un modèle qui traite la paire (requête, document) conjointement pour produire un score de pertinence plus fin que la simple similarité cosinus [127]. Dans les systèmes agentiques personnalisés, ce reclassement intègre également le profil utilisateur : les documents correspondant aux centres d'intérêt connus de l'utilisateur reçoivent un bonus de pertinence situationnelle [117].

Certains systèmes utilisent directement le LLM comme juge de pertinence (LLM-as-judge) [128] : le modèle évalue la pertinence de chaque passage par rapport à la requête et au profil utilisateur, produisant un score plus nuancé qu'un simple cross-encoder.

• **Composant de génération augmentée (RAG) :**
Les documents reclassés et filtrés sont intégrés dans le prompt du LLM générateur, qui produit une réponse synthétique citant explicitement ses sources [119]. La qualité de cette génération dépend de :
(1) la longueur et la densité informationnelle du contexte fourni .
(2) la capacité du LLM à synthétiser des informations potentiellement contradictoires .
(3) les instructions de personnalisation intégrées dans le system prompt (niveau de détail, style, langue).

• **Composant de mémoire et de profil utilisateur :**
L'agent maintient une représentation persistante du profil utilisateur [118], structurée selon les dimensions identifiées au §1.4 : centres d'intérêt, historique de recherche, niveau d'expertise, contexte situationnel. Cette mémoire est mise à jour après chaque session et influence toutes les couches précédentes de la reformulation de la requête au reranking des résultats.

8.3 Retrieval itératif et recherche multi-étapes

L'une des contributions les plus significatives des agents LLM à la recherche d'information est la capacité à effectuer une recherche itérative multi-étapes [116, 4]. Contrairement aux SRI classiques qui effectuent une seule passe de retrieval, un agent peut :

— Émettre une première requête et analyser les résultats pour identifier les lacunes informationnelles ;

— Formuler des requêtes de suivi ciblant précisément ces lacunes ;

— Croiser des informations issues de plusieurs sources pour résoudre des contradictions ;

— Décider quand le niveau de confiance dans la réponse est suffisant pour arrêter la recherche.

IRCoT (Interleaving Retrieval with Chain-of-Thought) [129] entrelace les étapes de raisonnement et de retrieval : chaque nouvelle pensée du LLM peut déclencher une nouvelle recherche, dont les résultats alimentent à leur tour le raisonnement. IterRetGen [130] itère entre génération et retrieval, utilisant la réponse intermédiaire comme requête enrichie pour l'étape suivante.

8.4 Personnalisation agentique de la recherche

L'intégration du profil utilisateur dans un système de RI basé sur des agents LLMs crée ce que l'on peut appeler un système de RI personnalisée augmenté par LLM. Ce système opère la synthèse des deux paradigmes développés dans ce mémoire : la modélisation de l'utilisateur (§1.3–1.5) et les capacités agentiques des LLMs [117, 131].

8.4.1 Intégration du profil dans le processus agentique

Le profil utilisateur intervient à trois niveaux dans la boucle agentique :

— **Au niveau du system prompt** : une représentation compacte du profil (centres d'intérêt, expertise, historique récent) est injectée dans les instructions système du LLM, orientant l'ensemble de ses décisions , reformulation, sélection d'outils, style de réponse .

— **Au niveau du retrieval** : les vecteurs de profil sont combinés avec le vecteur de requête pour biaiser la recherche vers les documents correspondant aux préférences connues de l'utilisateur (personalised dense retrieval) [132] .

— **Au niveau du reranking** : les scores de pertinence sont pondérés par un score de cohérence avec le profil, implémentant la pertinence cognitive et situationnelle définies par [133].

8.4.2 Mise à jour dynamique du profil par l'agent

Contrairement aux systèmes de modélisation de l'utilisateur classiques qui mettent à jour le profil par batch, un agent LLM peut mettre à jour le profil en temps réel au fil de la session. L'agent infère des signaux implicites (termes des requêtes successives, documents consultés, reformulations de l'utilisateur) et met à jour sa représentation interne du profil via des opérations de mémoire long terme [134]. Cette adaptation en ligne réalise le concept d'évolution du profil décrit au §1.4.6.

8.5 Navigation web agentique pour la RI ouverte

Pour les besoins informationnels qui dépassent les corpus pré-indexés, les agents LLM peuvent être équipés de capacités de navigation web autonome. WebGPT [135] a été l'un des premiers systèmes à entraîner un LLM à effectuer des recherches sur le web, à cliquer sur des liens, à lire des pages et à synthétiser les informations collectées. Des systèmes plus récents comme Perplexity AI et SearchGPT combinent retrieval en temps réel, navigation web et génération augmentée pour produire des réponses à jour avec sources citées.

Cette capacité est particulièrement précieuse dans les contextes où les corpus statiques sont insuffisants : actualité récente, données en temps réel (cours financiers, résultats sportifs, données scientifiques publiées après la date de cut-off du modèle), ou informations hautement spécifiques absentes des corpus d'entraînement.

8.6 Évaluation des systèmes de RI agentiques

L'évaluation des systèmes de recherche d'information (RI) augmentés par des agents basés sur les grands modèles de langage (LLMs) soulève des défis spécifiques qui dépassent les protocoles classiques décrits au §1.7 [120, 136]. Les métriques traditionnelles, telles que la précision, le rappel et le $nDCG$, s'avèrent nécessaires mais insuffisantes, car elles ne permettent pas de capturer la dimension générative et comportementale du système. Par conséquent, trois dimensions fondamentales et supplémentaires doivent être impérativement prises en compte pour auditer ces architectures :

— **Fidélité et attribution** :
La fidélité mesure le degré d'alignement entre les affirmations de la réponse générée et les documents sources effectivement récupérés, garantissant ainsi que le système n'introduit pas d'hallucinations factuelles absentes du corpus. Le cadre RAGAS [137]formalise cette évaluation automatisée des pipelines RAG à travers trois métriques complémentaires :

Faithfulness (Fidélité) : Évalue la proportion d'affirmations générées qui peuvent être logiquement déduites des passages récupérés. Son calcul repose sur l'extraction par le LLM des propositions vérifiables, suivie d'une vérification de leur présence sémantique dans le contexte.

Answer Relevance (Pertinence de la réponse) : Quantifie la capacité du modèle à répondre directement à la question posée, en mesurant l'absence de digressions ou de redondances informationnelles.

Context Relevance (Pertinence du contexte) : Évalue la proportion de passages récupérés qui sont strictement pertinents pour la requête, servant ainsi d'indicateur de performance pour le module de retrieval.

Note méthodologique : Ces métriques sont évaluées de manière automatisée par le biais d'un LLM évaluateur (LLM-as-a-judge). Bien que cette approche offre une excellente scalabilité, elle demeure dépendante des capacités cognitives du modèle pivot. En pratique, l'application de ce cadre méthodologique est fréquemment couplée à des études utilisateurs empiriques afin de consolider la validation de la fidélité dans des environnements hautement personnalisés.

— **Efficacité agentique** : L'évaluation de l'efficacité agentique s'attache à mesurer la qualité du processus computationnel de résolution et la trajectoire de raisonnement, au-delà du simple résultat final :

Nombre d'étapes de retrieval : Un agent optimal doit converger rapidement vers l'information critique ; un nombre disproportionné d'itérations traduit soit un raisonnement inefficace, soit une ambiguïté structurelle de la requête.

Cohérence de la trajectoire : Analyse si la succession des décisions prises par l'agent (actions et pensées) forme un enchaînement logique, progressif et téléologique vers la résolution du besoin informationnel.

Taux d'appels d'outils redondants : Calcule la proportion d'interactions avec des API ou des outils externes dont les résultats n'ont pas contribué à la synthèse finale, servant d'indicateur de sur-utilisation des ressources et d'évitement des appels inutiles.

Latence de bout en bout (End-to-End Latency) : Temps total de traitement requis depuis l'émission de la requête jusqu'à la génération finale, englobant l'ensemble des cycles de réflexion et d'action de l'agent.

— **Personnalisation et satisfaction utilisateur** : Les métriques de personnalisation quantifient si les résultats sont mieux adaptés au profil spécifique et au contexte de l'utilisateur que

ceux d'un système standard non personnalisé, une dimension absente des campagnes d'évaluation classiques [117] :nDCG@k personnalisé : Extension du $nDCG$ standard où les gains de pertinence sont pondérés dynamiquement par le score de cohérence avec le profil de l'utilisateur, substituant ainsi une approche situationnelle aux évaluations thématiques binaires.Pertinence Relative (RR) : Évalue la convergence entre la pertinence algorithmique et la pertinence contextuelle objective via le calcul de la similarité cosinus entre les vecteurs de jugements respectifs.Satisfaction utilisateur : Mesurée par le biais d'analyses empiriques et d'études utilisateurs dédiées évaluant la satisfaction perçue, le niveau de confiance accordé aux sources et la charge cognitive imposée. Ces études restent indispensables pour valider l'utilité réelle de la personnalisation.Cohérence de la personnalisation : Évalue l'indice de divergence des résultats lorsque deux profils distincts soumettent une requête identique, validant ainsi la sensibilité du système aux particularités de l'utilisateur.

8.7 Défis ouverts et perspectives

Malgré les progrès remarquables accomplis, plusieurs défis restent ouverts pour les agents LLM appliqués à la RI [113] :

- **Scalabilité** : les pipelines agentiques multi-étapes sont significativement plus coûteux que la recherche BM25 classique. Des approches de cache et de retrieval adaptatif sont nécessaires pour les déploiements à grande échelle ;
- **Robustesse aux sources adversariales** : les agents qui naviguent le web sont susceptibles d'intégrer des contenus manipulatoires ou incorrects (prompt injection, empoisonnement de résultats) .
- **Confidentialité et consentement** : la construction automatique de profils utilisateurs détaillés soulève des questions éthiques majeures au regard des réglementations (RGPD en Europe) .
- **Évaluation standardisée** : l'absence de benchmarks unifiés pour les systèmes RAG personnalisés freine la comparaison objective des approches ;
- **Explicabilité** : les décisions de l'agent ,pourquoi telle requête a été émise, pourquoi tel document a été sélectionné , restent difficiles à expliquer à l'utilisateur final.

9 Retrieval-Augmented Generation (RAG)

9.1 Principe et motivation

La Retrieval-Augmented Generation (RAG), introduite par Lewis et al. [119], est une architecture hybrde qui combine la puissance générative des LLMs avec l'accès dynamique à des bases de connaissances externes. L'idée fondamentale consiste à conditionner la génération du LLM non seulement sur la requête de l'utilisateur, mais aussi sur un ensemble de documents pertinents récupérés à la volée depuis une base documentaire indexée.

La RAG répond à trois limitations majeures des LLMs seuls :
(1) les connaissances obsolètes, bornées par la date de cut-off de l'entraînement .
(2) les hallucinations factuelles, en ancrant la génération dans des sources vérifiables .
(3) le manque de transparence, les documents sources pouvant être cités et vérifiés par l'utilisateur [112].

9.2 Architecture d'un pipeline RAG

Un pipeline RAG standard comprend cinq étapes :

1. **Indexation** : les documents sources sont découpés en chunks, encodés en vecteurs denses par un modèle d'embedding (ex. : E5, BGE, text-embedding-ada-002), et stockés dans une base vectorielle (FAISS, Chroma, Weaviate, Milvus) .
2. **Encodage de la requête** : la requête utilisateur est encodée dans le même espace vectoriel que les documents .
3. **Retrieval** : les k documents les plus similaires à la requête sont récupérés par recherche approximative des plus proches voisins (ANN) .
4. **Reranking (optionnel)** : un cross-encoder reclasse les candidats pour affiner la sélection [127] .
5. **Génération** : les documents récupérés sont concaténés avec la requête dans le prompt du LLM, qui génère une réponse synthétique et sourcée.

9.3 Variantes avancées : RAG modulaire et RAG agentique

La RAG standard a évolué vers des architectures plus sophistiquées. La RAG modulaire [138] décompose le pipeline en modules spécialisés et interchangeables : recherche itérative, génération de sous-requêtes, fusion de résultats. La RAG agentique confie à un agent LLM le contrôle de la stratégie de retrieval : l'agent décide quand chercher, formule les requêtes, évalue la pertinence des résultats, et itère si nécessaire. Self-RAG [4] pousse cette logique en entraînant le LLM à auto-évaluer la pertinence des passages récupérés et la fiabilité de ses propres générations via des tokens de réflexion spéciaux.

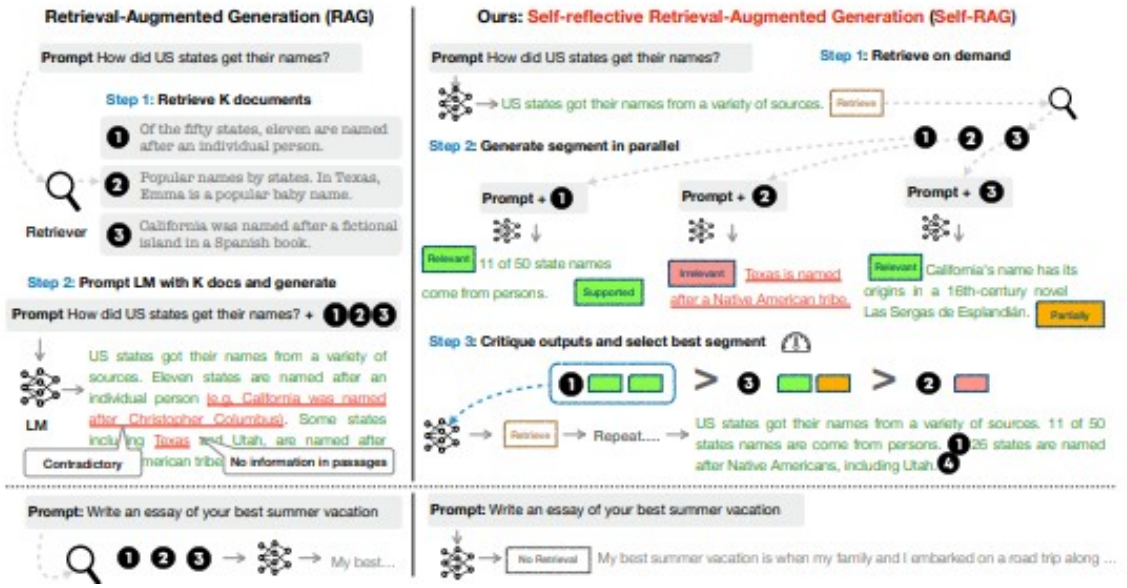


FIGURE II.5 – Présentation de SELF-RAG. SELF-RAG apprend à extraire, analyser et générer des passages de texte afin d'améliorer la qualité globale, la véracité et la vérifiabilité des textes générés.[4]

10 Conclusion

Les Large Language Models (LLMs) représentent une avancée majeure dans le domaine de l'intelligence artificielle et du traitement automatique du langage naturel. Leur développement a profondément transformé les systèmes de recherche et de génération de texte, en leur permettant de comprendre et de produire du langage naturel avec un niveau élevé de précision et de flexibilité.

Dans ce chapitre, nous avons d'abord présenté la définition des LLMs ainsi que leurs principes de fonctionnement basés sur l'architecture Transformer. Ensuite, nous avons décrit leur évolution historique, depuis les approches symboliques et statistiques jusqu'aux modèles neuronaux modernes à grande échelle.

Nous avons également étudié les différentes étapes de construction des LLMs, notamment le pré-entraînement sur de vastes corpus de données, ainsi que les techniques d'alignement comme le RLHF et les méthodes d'adaptation efficaces telles que LoRA. Par la suite, nous avons mis en évidence les principales capacités des LLMs, en particulier l'apprentissage en contexte, le raisonnement et la génération de réponses.

Enfin, nous avons abordé les limites de ces modèles, notamment le problème des hallucinations, ainsi que les solutions proposées pour améliorer leur fiabilité, comme les architectures basées sur le Retrieval-Augmented Generation (RAG). Nous avons aussi présenté l'émergence des agents LLM et leur rôle dans l'évolution des systèmes de recherche d'information vers des approches plus intelligentes et personnalisées.Enfin, ce chapitre constitue une base théorique essentielle pour l'intégration des LLMs dans les systèmes de recherche d'information personnalisée, objectif central de ce travail.

Chapitre III :

Conception du Système RIP-LLM

1 Introduction

Ce chapitre, qui représente notre contribution, traite un des sujets principaux de la recherche d'information personnalisée. Il s'agit de la reformulation et l'expansion de la requête initiale de l'utilisateur dans l'objectif de minimiser sa désambigüisé, et de bien cibler la recherche, et par conséquent améliorer la pertinence des résultats de recherche.

Notre contribution se résume à la combinaison de la personnalisation et l'exploitation des LLMs. Nous voulons profiter des capacités linguistiques intelligentes des LLMs tout en restant aligné avec les intérêts de l'utilisateur.

Afin d'assurer une reformulation contextuelle intelligente et non abusive, nous proposons une approche de personnalisation basée sur les étapes clés suivantes :

— Modélisation de l'utilisateur : l'utilisateur est modélisé via son profil qui regroupe les données statiques introduite explicitement par l'utilisateur (nom, prénom, age, pays, fonction, domaines d'intérêts, préférences, . . .) et les données dynamiques extraites implicitement de son historique de recherche (requêtes précédentes, document pertinents, termes pertinents).

— Préparation du contexte courant de la recherche : souvent, les nouvelles recherches de l'utilisateur sont en relation avec les recherches précédentes. C'est pourquoi nous utilisons un premier Agent-LLM pour préparer ce que nous appelons « Préparation de l'historique récen » pour le fournir au deuxième LLM qui a pour objectif de reformuler la nouvelle requête de l'utilisateur. Le contexte courant se compose de 20 dernières requêtes comme historique récent, et 50 termes les plus pondérés comme historique général.

— Reformulation intelligente et interactive de la requête d'utilisateur : sur la base du contexte fourni par le premier Agent-LLM, le deuxième LLM propose un ensemble de reformulations de la requête utilisateur. Ce dernier pourra choisir l'une des reformulations proposées comme il peut garder sa requête initiale en cas où les reformulations ne lui plaisent pas.

— Mis à jour du profil d'utilisateur : après chaque cession de recherche, les documents jugés pertinents par l'utilisateur sont fournis au troisième Agent-LLM pour les traiter et extraire les termes pertinents. Ces termes avec leurs documents, et les requêtes envoyées sont sauvegardés dans le profil utilisateur.

Dans le reste du chapitre, ces points sont expliqué en détail.

2 Profil Utilisateur

Le **Profil utilisateur** est une représentation structurée des caractéristiques, des centres d'intérêt, des préférences et des besoins informationnels d'un utilisateur. Il est construit à partir de données explicites fournies par l'utilisateur ainsi que de données implicites issues de ses interactions avec le système, notamment son historique de recherche.

Dans notre système de la recherche d'information personnalisée, le profil utilisateur intègre les centres d'intérêt, les préférences, la localisation, l'historique des requêtes, les documents pertinent et les termes extrait a partir les documents pertinent. Ces informations sont exploitées pour identifier le contexte de recherche de l'utilisateur et personnaliser la reformulation des requêtes afin d'améliorer la pertinence des résultats retournés.

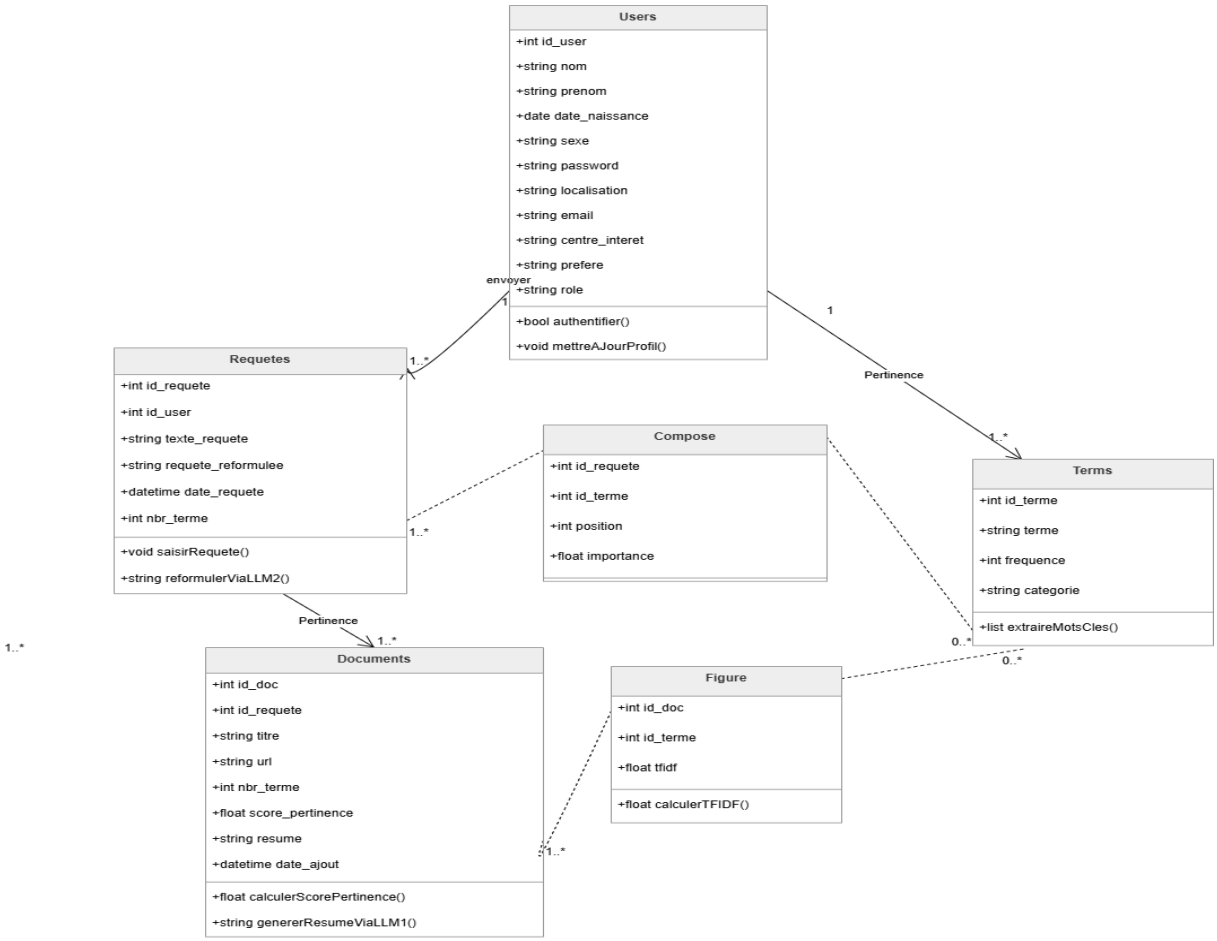


FIGURE III.1 – Architecture de données : Diagramme de classes UML

2.1 Partie statique

La partie statique du profil utilisateur regroupe l'ensemble des informations relativement stables qui permettent d'identifier et de caractériser l'utilisateur au sein du système de recherche d'information. Ces données sont généralement renseignées lors de l'inscription et évoluent rarement au cours du temps.

En a représentée par la table **users**, qui contient notamment l'identifiant de l'utilisateur **id-user**, le nom, le prénom, la date de naissance, le sexe, l'adresse électronique,le mot de passe , la localisation, ainsi que les centres d'intérêt et les préférences générales. Ces informations constituent le profil de base de l'utilisateur et fournissent les premières indications nécessaires à la personnalisation du système.

Ainsi, la partie statique joue un rôle fondamental dans l'initialisation du profil utilisateur en fournissant une base de connaissances qui sera progressivement enrichie par les informations dynamiques issues des interactions et des activités de recherche de l'utilisateur.

2.2 Partie dynamique

La partie dynamique du profil utilisateur regroupe l'ensemble des informations évolutives construites à partir des interactions de l'utilisateur avec le système de recherche d'information personalisee. Contrairement à la partie statique, elle évolue continuellement en fonction des requêtes soumises, des documents consultés et des termes extraits. Elle constitue le cœur du

processus de personnalisation puisqu'elle permet de représenter les intérêts réels de l'utilisateur à partir de son comportement de recherche.

• **Historique de Recherche Pertinent** Cette composante (figure3.1) sert pour sauvegarder les traces des activités de recherche les plus significatives pour l'‘utilisateur. Elle se compose des 3 sous-ensembles suivants :

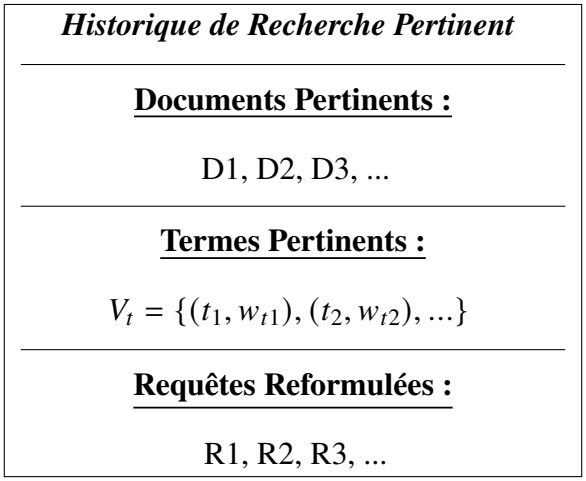


FIGURE III.2 – Historique de Recherche Pertinent de l'Utilisateur

- Requêtes Reformulées :

Les requêtes reformulées correspondent aux nouvelles versions des requêtes initialement saisies par l'utilisateur. Elles sont générées par le système à partir de l'analyse de la requête originale et des informations contenues dans le profil utilisateur. L'objectif de cette reformulation est d'enrichir la requête avec des termes plus représentatifs du besoin informationnel exprimé et du contexte de recherche de l'utilisateur.

- Les documents pertinents :

Les documents pertinents correspondent aux documents que le système identifie comme les plus susceptibles de satisfaire le besoin informationnel de l'utilisateur. Ils sont obtenus à partir des résultats de recherche associés aux requêtes formulées et constituent une source essentielle pour la construction et l'enrichissement du profil utilisateur. L'analyse de ces documents permet d'extraire les termes les plus représentatifs des centres d'intérêt de l'utilisateur et d'actualiser son profil dynamique. Ainsi, les documents pertinents jouent un rôle fondamental dans la personnalisation de la recherche, en contribuant à l'amélioration de la pertinence des résultats retournés ainsi qu'à la génération de reformulations adaptées au contexte de recherche.

- Les termes pertinents :

Les termes pertinents sont les mots les plus représentatifs extraits des documents consultés par l'utilisateur. Ils permettent d'identifier ses centres d'intérêt et les thèmes qui caractérisent son profil de recherche.

Pour mesurer leur importance, nous utilisons la ponderation TF-IDF, qui attribue un poids à chaque terme selon :

-sa fréquence d'apparition dans les documents de l'utilisateur (TF) .

-sa rareté dans l'ensemble des document de profil (IDF).

$$Poids = \log(1 + (\frac{tf}{maxfrequence})) \times \log\left(\frac{N}{n}\right)$$

Avec :

— N : le nombre total de documents de lhistorique .

— n : le nombre de documents contenant le terme .

— tf : fréquence totale de terme t dans les documents de l'historique.

— $maxfrequence$: la plus grande fréquence des termes dans les documents de l'historique.

- Relation requêtes–termes (Composer) :

La relation requêtes–termes (Composer) représente l'association entre une requête formulée par l'utilisateur et les différents termes qui la constituent.
Avant leur stockage dans cette relation, les requêtes subissent plusieurs étapes de prétraitement, Elles sont d'abord nettoyées afin d'éliminer les caractères spéciaux, la ponctuation et les éléments non pertinents, puis tokenisées pour être découpées en unités lexicales (tokens).Ensuite, les mots vides (stopwords) sont supprimés afin de ne conserver que les termes porteurs de sens. Une fois ce traitement effectué, chaque requête est décomposée en un ensemble de mots-clés significatifs reflétant le besoin informationnel de l'utilisateur.

Le rôle de cette relation est de facilite l'analyse du contenu des requêtes, l'identification des termes les plus représentatifs ainsi que la construction et l'enrichissement du profil utilisateur. De plus, l'attribut position permet de conserver l'ordre d'apparition des termes dans la requête, information pouvant être exploitée lors des processus de personnalisation et de reformulation des requêtes.

- Relation document–termes (Figure) :

La relation document–termes (Figure) établit l'association entre les documents et les termes qui y apparaissent. Après les étapes de prétraitement du texte, notamment le nettoyage, la tokenisation et la suppression des mots vides, les termes significatifs sont extraits de chaque document. Cette relation permet de représenter le contenu documentaire sous forme d'un ensemble de termes descriptifs facilitant son analyse et son exploitation dans le système de recherche d'information.

Elle contient également l'attribut de pondération **TF-IDF**, qui permet d'évaluer l'importance de chaque terme dans un document donné. Le score TF-IDF mesure le caractère représentatif d'un terme en tenant compte à la fois de sa fréquence dans le document et de sa rareté dans l'ensemble de la collection documentaire.

Cette information est utilisée pour identifier les termes les plus pertinents et mieux caractériser le contenu des documents afin de faciliter leur exploitation dans les mécanismes de recherche et de recommandation.

- Relation Users–termes (Pertinence) :

Cette relation permet de représenter le profil d'un utilisateur à travers les termes les plus significatifs extraits des documents pertinents . Après les étapes de prétraitement et ponderation, chaque terme est évalué afin de mesurer son importance dans les centres d'intérêt de l'utilisateur.

L'évaluation de la pertinence repose sur les attributs suivants :

tf (Term Frequency) : fréquence d'apparition du terme dans les documents pertinents de l'utilisateur.

Poids : mesure l'importance du terme dans le profil utilisateur en combinant sa fréquence et sa présence dans les documents pertinents.

Un terme ayant une fréquence élevée et apparaissant dans plusieurs documents pertinents est considéré comme plus représentatif des intérêts de l'utilisateur. Les termes possédant les poids les plus importants contribuent ainsi à caractériser son profil de manière plus précise.

Les informations de cette relation sont stockées dans la base de pertinence utilisateur–termes afin de maintenir une représentation dynamique des centres d'intérêt. Elles sont ensuite exploitées pour enrichir le profil utilisateur, reformuler les requêtes, améliorer les résultats de recherche et générer des recommandations personnalisées.

3 Processus de Recherche

Le processus de recherche proposé dans RIP-LLM a pour objectif d'améliorer la pertinence des résultats retournés à l'utilisateur en exploitant les informations stockées dans son profil ainsi que les capacités des Grands Modèles de Langage (LLM). Contrairement à une recherche classique qui repose uniquement sur la requête saisie, notre approche intègre les connaissances acquises à partir des recherches précédentes afin de mieux comprendre le besoin informationnel de l'utilisateur.

Le processus de recherche s'effectue à travers un ensemble d'étapes successives, chacune contribuant à la personnalisation de la recherche et à l'amélioration de la qualité des résultats proposés.

3.1 Préparation de l'historique récent (Agent-LLM1)

le figure suivant represente le prompt de AGENT-LLM1 :

```
prompt = f"""
    Tu es le Agent-LLM1 de preparation du profil utilisateur.

    a partir des donnees suivantes, construis un contexte utilisateur
    clair pour un systeme de recherche personalisee.

    Donnees :

    {json.dumps(raw_context, ensure_ascii=False, indent=2, default=str)}

    Taches :
    1. Garder le profil utilisateur.
    2. Garder l'historique des 15 dernieres requetes.
    3. Garder les 50 termes pertinents avec TF-IDF.
    4. Ne pas inventer de donnees.
    5. Ne pas modifier les valeurs TF-IDF.
    6. Resumer les interets dominants de l'utilisateur.
    7. Repondre uniquement en JSON valide.

    Format obligatoire :
    {
        "agent": "LLM1 - Agent Prepare",
        "date_update": "{datetime.now().strftime('%Y-%m-%d %H:%M:%S')}",
        "user": {},
        "historique": [],
        "termes": [[],
        "resume_profil":  "",
        "themes_dominants": ([]
    }
"""
```

FIGURE III.3 – Prompt de AGENT-LLM1

La première étape du processus de recherche personnalisée est assurée par l'agent-LLM1, dont la mission consiste à construire un contexte utilisateur exploitable pour la reformulation des requêtes.

Premièrement, lorsque l'utilisateur accède à son espace de recherche, l'agent-LLM1 est automatiquement déclenché afin de préparer les informations contextuelles nécessaires aux étapes suivantes.

Deuxièmement, l'agent-LLM1 interroge la base de données et récupère les éléments les plus représentatifs du profil dynamique de l'utilisateur, notamment les centres d'intérêt, les préférences générales, la localisation et**les dernières requêtes reformulées ainsi que les N termes les plus pertinents** caractérisant ses centres d'intérêt.

Troisièmement, ces informations sont sélectionnées, organisées et structurées dans un fichier au format **JSON**, conçu pour fournir une représentation synthétique et actualisée du contexte utilisateur.

Quatrièmement, le fichier généré est mis à la disposition de **LLM2** pour les étapes suivantes du processus.

Par ailleurs, la mise à jour de l'agent-LLM1 s'effectue après les recherches de l'utilisateur. Il conserve les requêtes et les termes les plus importantes afin de maintenir un profil utilisateur pertinent et à jour.

Enfin,L'objectif principal de Agent-LLM1 est de préparer à l'avance les données nécessaires à l'analyse des requêtes. Cette approche permet à LLM2 de disposer directement d'un contexte prêt à l'emploi sans devoir interroger la base de données à chaque nouvelle recherche. Ainsi, le temps de traitement est réduit, les accès répétés à la base de données sont évités et le processus de reformulation gagne en efficacité.

Le fichier JSON généré par LLM1 sert de mémoire contextuelle temporaire, offrant à LLM2 un accès rapide aux informations nécessaires et facilitant ainsi le traitement des requêtes.

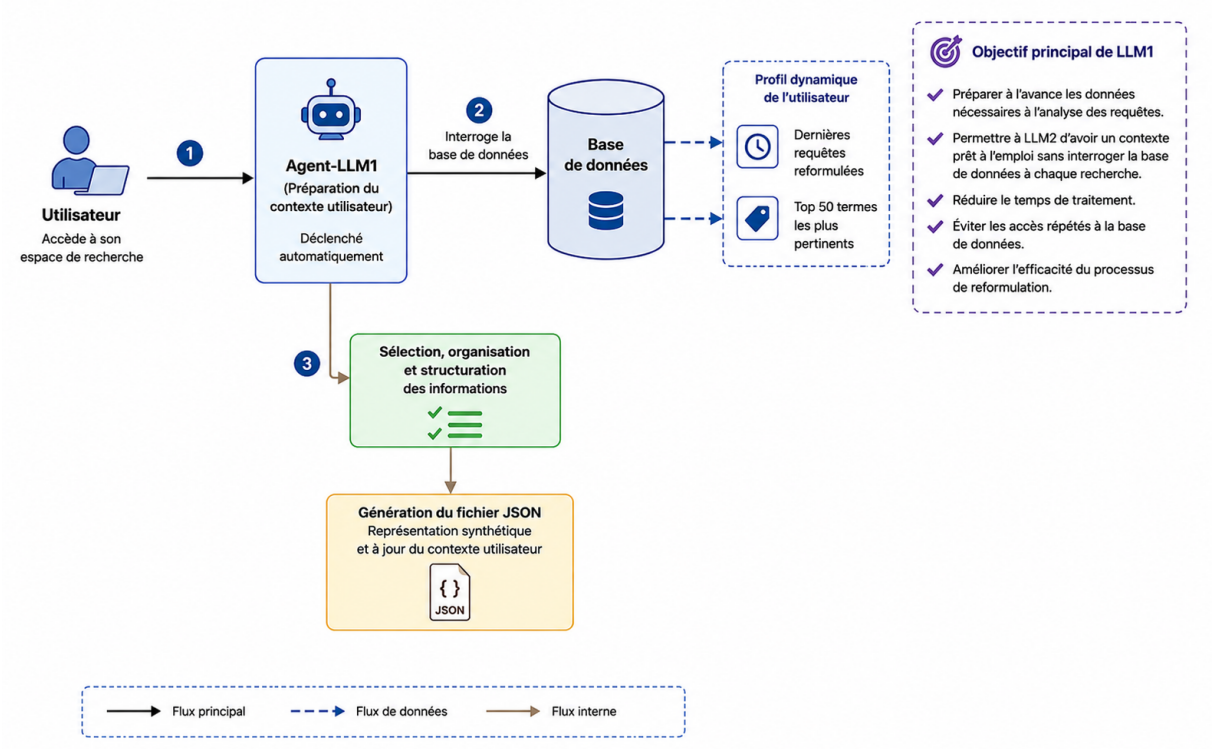


FIGURE III.4 – AGENT-LLM1 : Prépare le contexte utilisateur a partir de son profil

3.2 Reformulation de la requête (LLM2)

Après la phase de préparation du contexte utilisateur réalisée par l'agent-LLM1, **l'agent-LLM2** intervient pour effectuer la reformulation de la requête soumise par l'utilisateur.

Premièrement, l'agent-LLM2 charge le fichier JSON généré par l'agent-LLM1. Ce fichier contient les informations essentielles du profil utilisateur.

Deuxièmement,l'agent analyse la requête courante et détermine son type de contexte. Il vérifie si la requête est liée avec l'historique de l'utilisateur. Si aucun lien pertinent n'est trouvé, la requête est considérée comme générale.

Et selon le contexte détecté, l'agent choisit le prompt approprié.Si la requête est liée au profil utilisateur, il exploite le contexte disponible afin de produire des reformulations personnalisées et plus précises, Sinon en cas de la requête est générale, il génère des reformulations proches du sujet initial sans introduire de nouveau thème.

Troisièmement, l'agent envoie le prompt au modèle Gemini afin de générer cinq reformulations optimisées sous forme JSON.
La figure ci-dessous illustre le prompt de l'agent LLM2 en cas **generale** :

```
context_type = detect_query_context(
    query,
    termes,
    user,
    historique)

print("contexte =", context_type)

if context_type == "general":
    prompt = f"""
    Tu es un expert en recherche d'information.

    PROFIL UTILISATEUR:
    - Localisation: {user.get('localisation','')}
    - Centre d'intérêt: {user .get('centre_interet','')}
    - Préférences: {user.get('prefere','')}

    Requête utilisateur:
    {query}

    Objectif:
    Produire 5 reformulations proches du sujet initial.

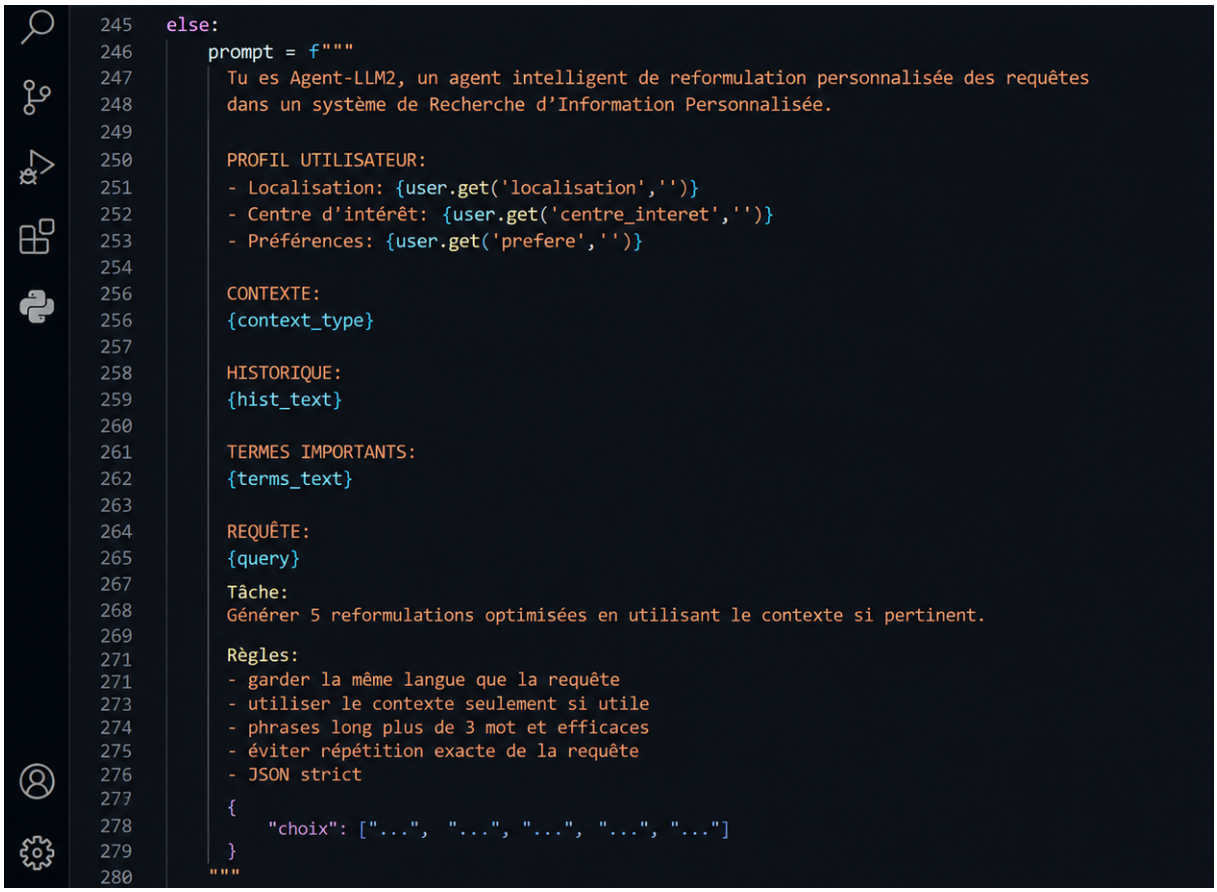
    Règles:
    - rester dans le même thème
    - ne pas changer le sens principal
    - ne pas introduire un nouveau sujet
    - reformulations adaptées à un moteur de recherche
    - phrases long plus de 3 mot et efficaces
    - éviter répétition exacte de la requête
    - utilise le profil de utilisateur

    Réponds uniquement en JSON:
    {
       "choix":[
        "...",
        "...",
        "...",
        "..."
    ]
"""
```

Python 3.10.9 64-bit 0 0 Ln 200, Col 1 Spaces: 4 UTF-8 CRLF

FIGURE III.5 – Prompt de AGENT-LLM2(cas :generale)

La figure ci-dessous illustre le prompt de l'agent LLM2 en cas **Specifique** :



```
else:
    prompt = f"""
      Tu es Agent-LLM2, un agent intelligent de reformulation personnalisée des requêtes
      dans un système de Recherche d'Information Personnalisée.

      PROFIL UTILISATEUR:
      - Localisation: {user.get('localisation','')}
      - Centre d'intérêt: {user.get('centre_interet','')}
      - Préférences: {user.get('prefere','')}

      CONTEXTE:
      {context_type}

      HISTORIQUE:
      {hist_text}

      TERMES IMPORTANTS:
      {terms_text}

      REQUÊTE:
      {query}
      Tâche:
      Générer 5 reformulations optimisées en utilisant le contexte si pertinent.

      Règles:
      - garder la même langue que la requête
      - utiliser le contexte seulement si utile
      - phrases long plus de 3 mot et efficaces
      - éviter répétition exacte de la requête
      - JSON strict

      {
          "choix": ["...",  "...", "...",  "...", "..."]
      }
    """
```

FIGURE III.6 – Prompt de AGENT-LLM2

Quatrièmement, l'agent génère plusieurs reformulations candidates correspondant à différentes interprétations possibles du besoin informationnel exprimé. Ces reformulations sont ensuite présentées à l'utilisateur, qui sélectionne celle correspondant le mieux à son intention de recherche.

Enfin, la reformulation retenue est transmise au moteur de recherche afin d'obtenir les documents les plus pertinents.

L'objectif principal de l'agent-LLM2 est d'améliorer la qualité de la recherche en produisant des requêtes reformulées plus précises, plus représentatives du contexte de l'utilisateur et mieux adaptées à son besoin informationnel, tout en préservant le sens initial de la requête.

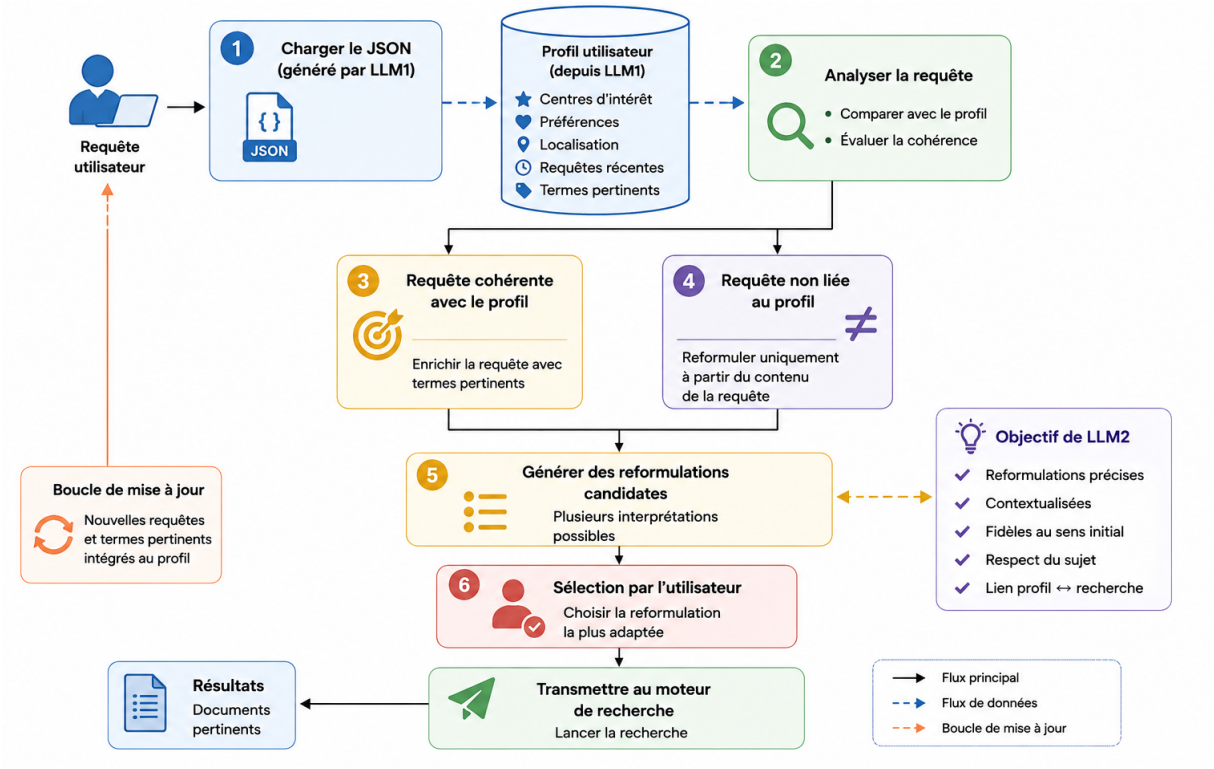


FIGURE III.7 – AGENT-LLM2 : Reformule de la requête utilisateur

3.3 Mise à jour de profil Agent-LLM3

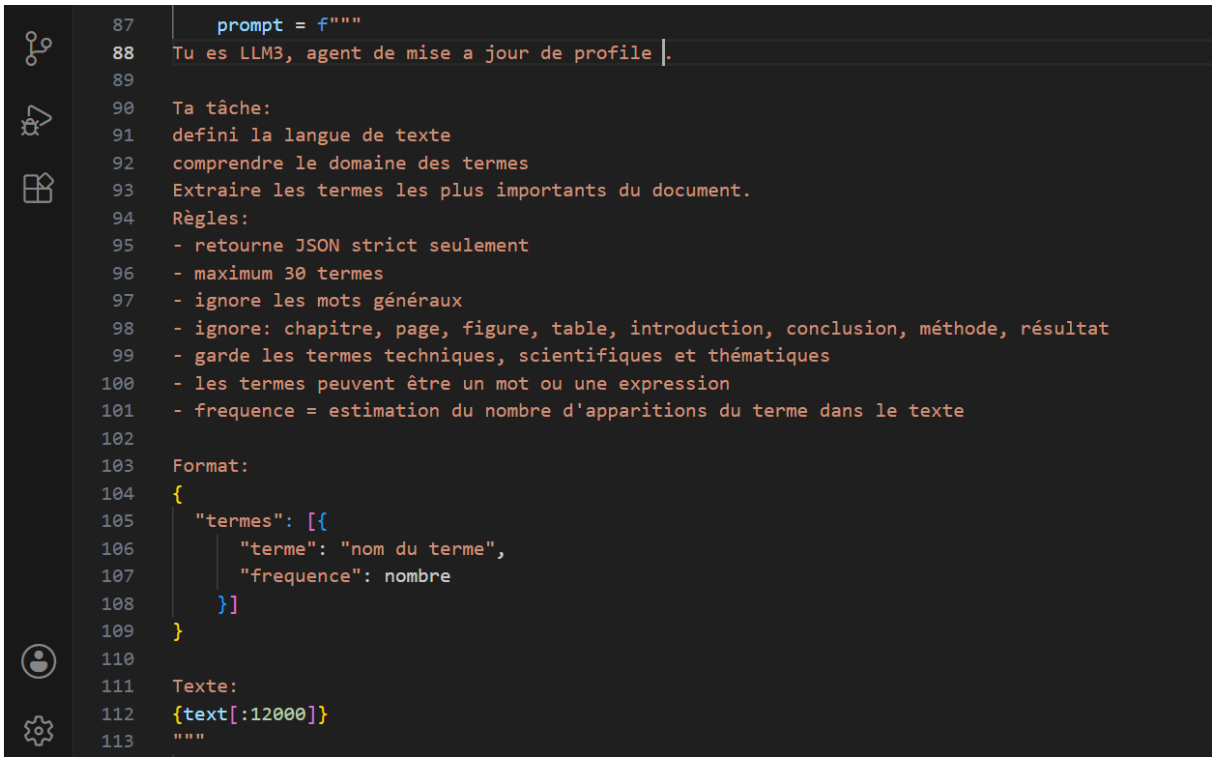
La dernière étape du processus de recherche est assurée par **l'agent-LLM3**, chargé de la mise à jour du profil utilisateur à partir des interactions réalisées avec les résultats de recherche.

Premièrement, l'agent récupère le document sélectionné par l'utilisateur à partir de son URL. puis extrait le texte des de pages.

Deuxièmement, le texte extrait est nettoyé afin de supprimer les liens, les espaces inutiles et les caractères non nécessaires.

Troisièmement, l'agent utilise le modèle Gemini pour analyser le contenu du document, détecter sa langue, comprendre son domaine et extraire les termes techniques, scientifiques ou thématiques les plus importants.

le figure suivant represente le prompt de AGENT-LLM3 :



```
    prompt = f"""
Tu es LLM3, agent de mise a jour de profile .

Ta tâche:
defini la langue de texte
comprendre le domaine des termes
Extraire les termes les plus importants du document.
Règles:
- retourne JSON strict seulement
- maximum 30 termes
- ignore les mots généraux
- ignore: chapitre, page, figure, table, introduction, conclusion, méthode, résultat
- garde les termes techniques, scientifiques et thématiques
- les termes peuvent être un mot ou une expression
- frequence = estimation du nombre d'apparitions du terme dans le texte

Format:
{
  "termes": [{
      "terme": "nom du terme",
      "frequence": nombre
    }]
}

Texte:
{text[:12000]}
"""
```

FIGURE III.8 – Prompt de AGENT-LLM3

Quatrièmement, les termes extraits sont normalisés, puis leurs fréquences sont calculées. L'agent calcule ensuite un poids de pertinence basé sur la fréquence du terme dans le document et son apparition dans les documents déjà consultés par l'utilisateur.

Enfin,pour chaque terme extrait, l'agent vérifie d'abord s'il existe déjà dans la base de données. Si le terme existe, ses informations de fréquence et de pertinence sont mises à jour. En revanche, si le terme n'existe pas, il est ajouté dans la table *termes*. Les relations et les poids associés sont ensuite enregistrés ou mis à jour dans les tables**users**, *figure*, *compose* et *pertinence* . Le profil utilisateur est également mis à jour à travers le nombre de documents consultés et la fréquence maximale des termes.

L'objectif principal de **l'agent-LLM3** est donc d'enrichir automatiquement le profil utilisateur à partir des documents consultés afin d'améliorer la personnalisation des recherches futures.



FIGURE III.9 – AGENT-LLM3 : Mis a jour du profil d'utilisateur

4 Environnement de développement

4.1 *Mouter de Recherche (Tavily)*

Tavily est une plateforme de recherche web basée sur une API permettant d'obtenir en temps réel des informations pertinentes issues du Web. Elle est particulièrement adaptée aux applications d'intelligence artificielle, aux agents conversationnels et aux systèmes de recherche d'information grâce à sa capacité à fournir rapidement des résultats actualisés.

Nous avons utilise Tavily comme moteur de recherche externe pour récupérer les documents correspondant qui pertinent pour l'utilisateur. Les résultats retournés, comprenant les titres, les liens et les résumés des documents, sont ensuite affichés dans l'interface du système afin d'améliorer l'accès à l'information et la pertinence des résultats proposés à l'utilisateur.

4.2 `Visual Studio Code VS`

Visual Studio Code (VS Code) est un éditeur de code source open source développé par Microsoft. Léger et extensible, il prend en charge de nombreux langages grâce à ses extensions.

Nous avons utilisé VS Code pour le développement du frontend et du backend, notamment avec les langages **Python**, **HTML** et **CSS**. Il a servi à l'édition du code, au débogage et à la gestion des versions avec Git. Ses fonctionnalités, telles que l'autocomplétion, la coloration syntaxique et les outils d'analyse du code, ont facilité le développement et amélioré la productivité.

4.3 API Gemini

L'**API Gemini** est une interface de programmation proposée par Google permettant d'accéder aux capacités des modèles de langage Gemini à travers des requêtes distantes. Nous avons utilisée pour réaliser des tâches de compréhension du contexte de profil LLM1, reformulation ddes requete LLM2, Mise a jour de profil LLM3. Son intégration facilite l'exploitation des fonctionnalités avancées de l'intelligence artificielle sans nécessiter l'installation ou l'exécution locale des modèles, tout en assurant une communication simple et efficace avec l'application.

4.4 Python

Python est un langage de programmation interprété de haut niveau, apprécié pour sa simplicité, sa lisibilité et la richesse de son écosystème logiciel. Il est largement utilisé dans le développement web, l'analyse de données, l'apprentissage automatique et l'intelligence artificielle.

Dans le cadre de notre projet, Python assure le développement des modules de traitement, la gestion des données et les services backend. Les principales bibliothèques exploitées sont :

- **Flask**

Flask est un micro-framework web basé sur Python, utilisé pour développer l'application web. Il assure la gestion des requêtes, la navigation entre les pages et la communication entre l'interface utilisateur et les composants backend, tout en offrant une architecture légère et flexible.

- **PyMySQL**

PyMySQL est une bibliothèque Python qui assure la connexion entre l'application et la base de données MySQL, permettant l'exécution des requêtes SQL et la gestion des données.

4.5 HTML / CSS

HTML et CSS sont utilisés pour la conception de l'interface utilisateur du système. HTML permet de structurer les différentes pages web (formulaires, boutons, zones de saisie).
le **CSS** assure la mise en forme et l'amélioration visuelle de l'interface.

4.6 MySQL

MySQL est un système de gestion de bases de données relationnelles (SGBDR) open source permettant de stocker de manière structurée les informations relatives aux utilisateurs, aux requêtes, aux documents et aux termes extraits, ainsi que d'organiser et de gérer efficacement ces données. Il assure l'intégrité, la cohérence et la fiabilité des informations grâce à son modèle relationnel. On a utilisé pour la gestion et la persistance des données nécessaires au fonctionnement de l'application.

4.7 XAMPP

XAMPP est un environnement de développement local utilisé pour exécuter et tester l'Application. Il fournit un serveur Apache ainsi qu'un système de gestion de base de données **MySQL**, facilitant ainsi le déploiement local de l'application.

Cet environnement permet de simuler un serveur réel et de vérifier le bon fonctionnement du système avant sa mise en production.

5 Interfaces du système

Les interfaces du système RIP-LLM constituent la couche de présentation permettant l'interaction directe entre l'utilisateur et les différents modules fonctionnels de l'application. Elles jouent un rôle essentiel dans la fluidité de l'utilisation, la compréhension des résultats et la navigation entre les fonctionnalités. L'ensemble des interfaces a été conçu de manière à garantir une ergonomie simple, intuitive et cohérente avec les objectifs du système de recherche personnalisée

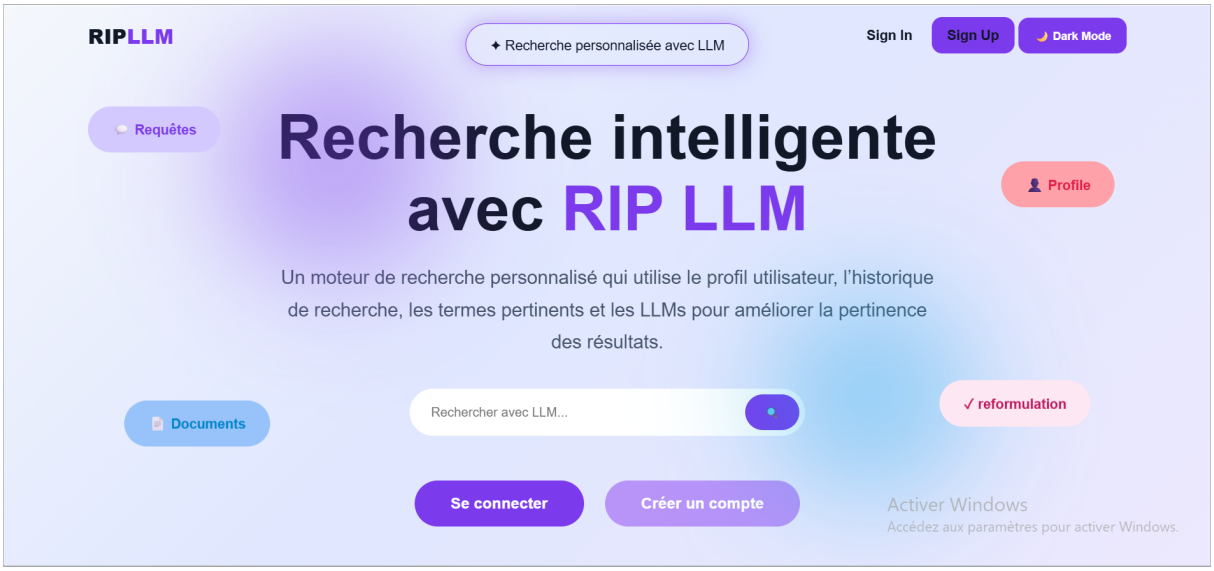


FIGURE III.10 – Interface du système

5.1 Interface d'authentification

L'interface d'authentification constitue le point d'entrée principal du système. Elle permet d'identifier chaque utilisateur à travers un formulaire sécurisé comprenant un identifiant et un mot de passe. Cette étape est indispensable pour assurer la personnalisation des résultats, car elle associe chaque session à un profil utilisateur spécifique.

Une fois l'authentification réussie, le système charge automatiquement les données associées à l'utilisateur, notamment son historique de recherche et son profil implicite. En cas d'échec, un message d'erreur est affiché afin d'informer l'utilisateur et de lui permettre de corriger ses informations.

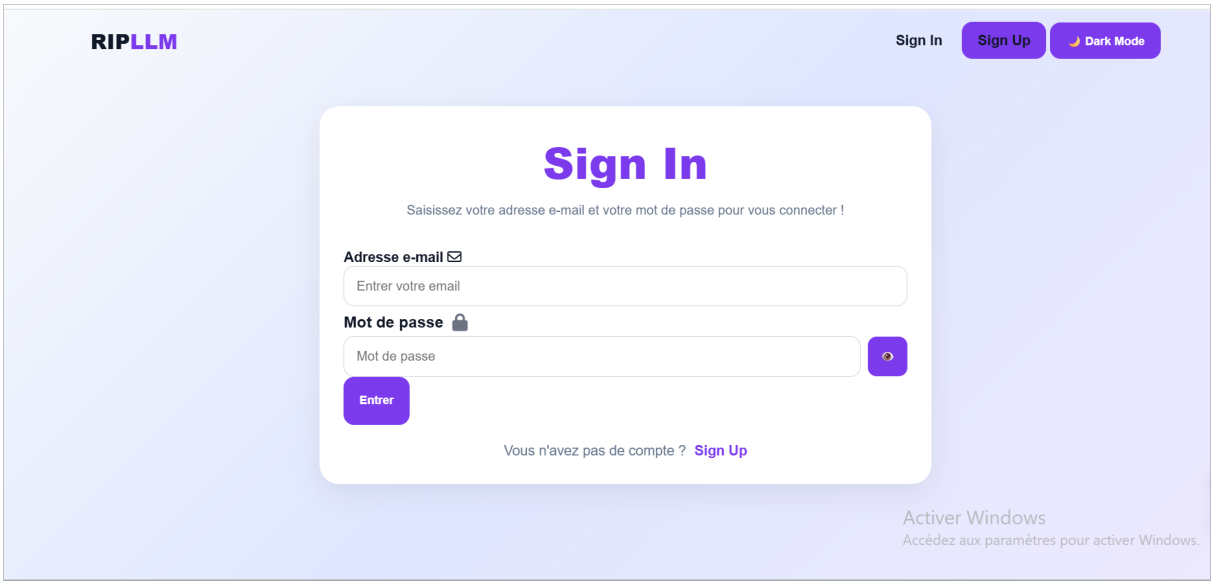


FIGURE III.11 – Interface de login

5.2 Interface de recherche

L'interface de recherche constitue le cœur functional du système. Elle permet à l'utilisateur de saisir ses requêtes sous forme textuelle dans un espace dédié, conçu de manière simple et épurée afin de faciliter la concentration sur l'action principale : la recherche d'information.

Une fois la requête soumise, elle est transmise au backend via Flask pour être traité par les différents modules du système, notamment le profilage utilisateur, l'analyse contextuelle et la reformulation.

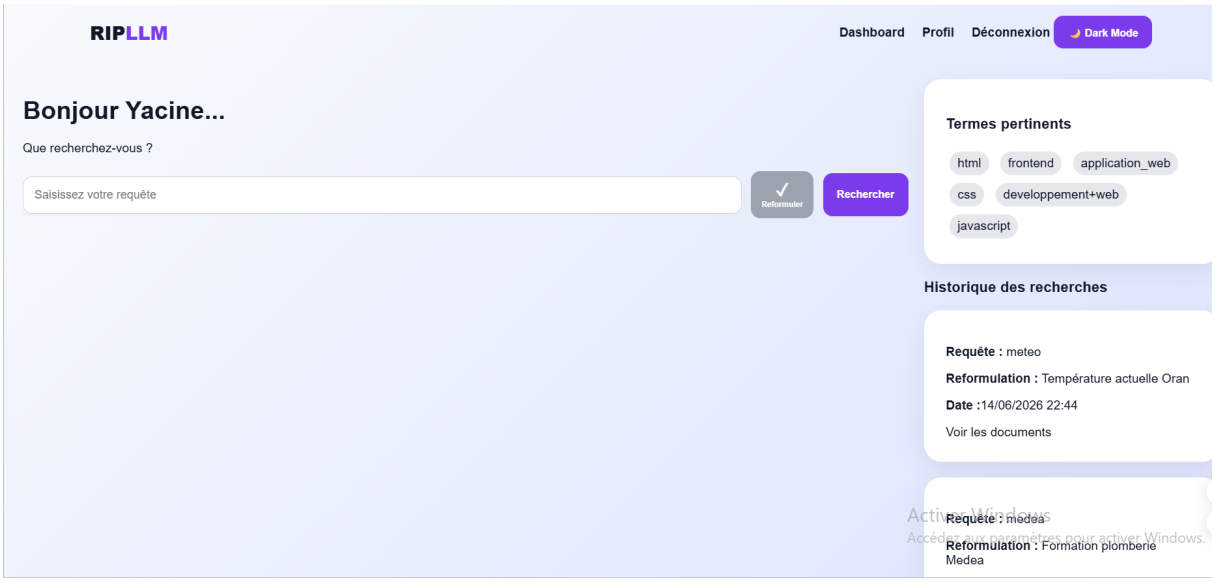


FIGURE III.12 – Saisie de la requete

5.3 Affichage des résultats

L'interface d'affichage des résultats présente les documents récupérés après traitement de la requête. Les résultats sont organisés de manière structurée afin de faciliter leur lecture et leur

exploitation par l'utilisateur.

Chaque résultat peut inclure un titre, un extrait pertinent ainsi que des informations complémentaires issues du processus de traitement. Le système met également en évidence certains termes clés ou correspondances sémantiques afin d'améliorer la compréhension globale.

L'objectif de cette interface est de fournir une réponse claire, pertinente et adaptée au contexte de recherche de l'utilisateur.

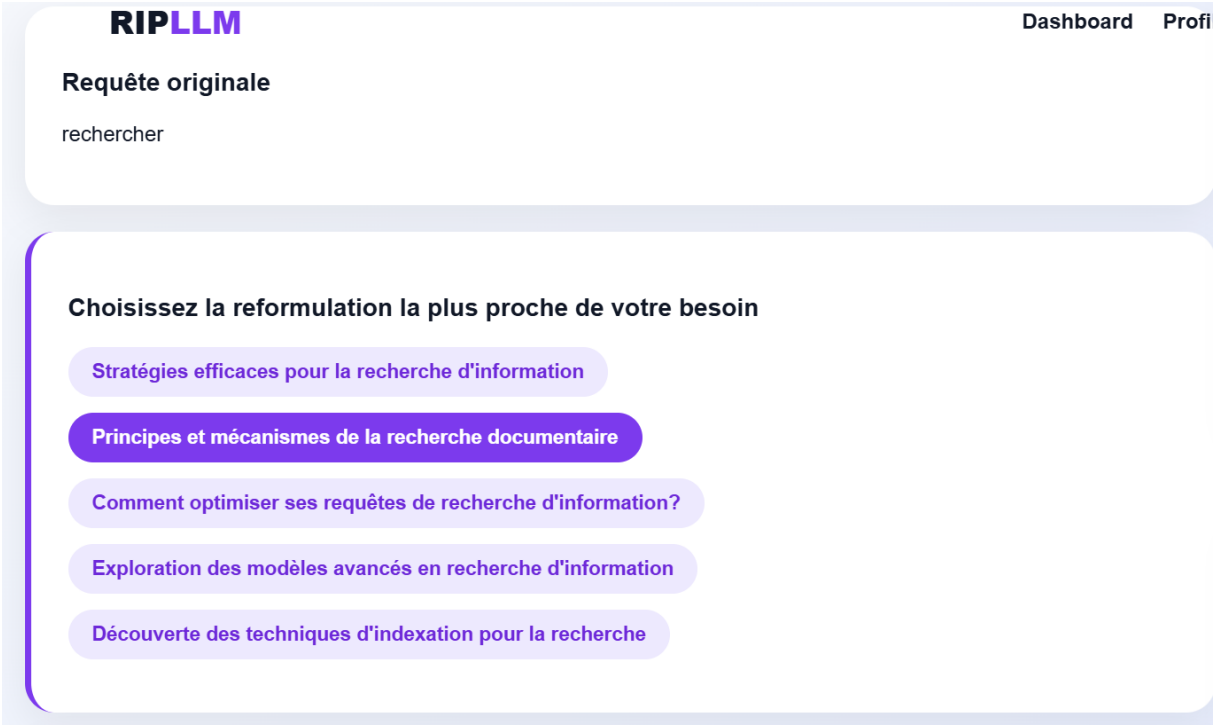


FIGURE III.13 – les reformulation de la requete



FIGURE III.14 – les reformulation de la requete

5.4 Consultation du profil utilisateur

Cette interface permet à l'utilisateur de consulter les informations relatives à son profil implicite. Elle présente les centres d'intérêt détectés par le système ainsi que les thèmes dominants extraits à partir de l'historique des recherches et des documents consultés.

Le profil est affiché de manière structurée afin de faciliter la compréhension de la représentation construite par le système. Cette transparence contribue à renforcer la confiance de l'utilisateur et à mieux expliquer le processus de personnalisation.

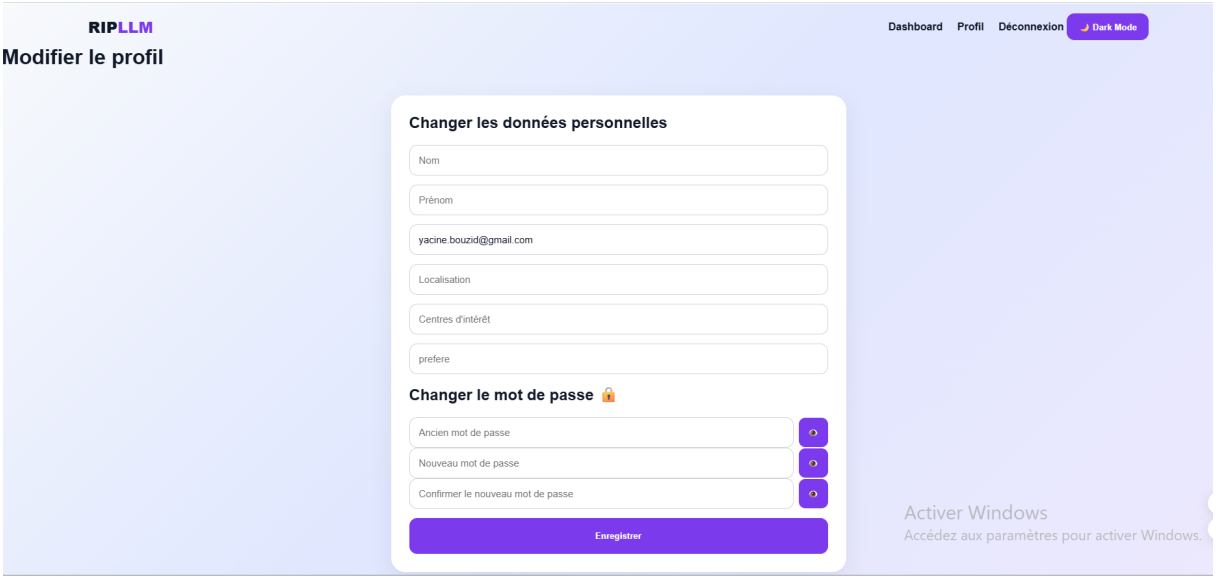


FIGURE III.15 – Interface de visualisation du profil sémantique implicite

6 Évaluation

6.1 Évaluation difficile dans le contexte de la recherche personnalisée

La pertinence dans le cadre de la recherche d'information personnalisée est fortement subjective [61], car elle dépend directement de l'utilisateur, qui demeure le seul capable de juger les résultats selon son propre point de vue et ses besoins informationnels. Cette caractéristique rend l'évaluation des systèmes de recherche d'information personnalisée particulièrement complexe. Dans ce contexte, un cadre d'évaluation des approches de recherche d'information personnalisée (RIP) repose principalement sur deux composantes essentielles [59] : les profils utilisateurs, qui regroupent des informations statiques et dynamiques décrivant les caractéristiques et les centres d'intérêt des utilisateurs, ainsi qu'une collection de tests composée d'un ensemble de documents, de requêtes et de jugements de pertinence fournis par les utilisateurs.

Cependant, en raison de l'absence de cadres d'évaluation standardisés spécifiquement dédiés à la personnalisation de la recherche, plusieurs travaux de recherche ont proposé leurs propres protocoles expérimentaux afin de pallier cette limite [59, 42, 41, 61].

6.2 Stratégie d'Évaluation

Dans l'absence d'un cadre d'évaluation standard dédié à la recherche d'information personnalisée, plusieurs approches proposent leurs propres protocoles expérimentaux. Dans ce travail, nous avons défini une stratégie d'évaluation structurée permettant d'analyser la performance de notre système selon différents scénarios.

Le protocole expérimental se déroule suivant les étapes systématiques décrites ci-dessous :

1. **Construction des profils utilisateurs :**
   Nous avons créé 10 profils représentant 10 utilisateurs issus de domaines professionnels variés, afin de garantir la diversité des centres d'intérêt. Chaque utilisateur dispose de son propre profil, construit en collaboration avec lui, reflétant ses préférences et son contexte informationnel.
2. **Besoins en information et requêtes :**
   Chaque utilisateur a formulé trois requêtes correspondant à ses besoins informationnels propres. Ces requêtes peuvent être spécifiques à l'utilisateur ou partagées entre plusieurs utilisateurs, afin d'évaluer la capacité du système à gérer différents types de besoins.
3. **Recherche des documents :**
   Les requêtes sont envoyées au moteur de recherche Google afin de récupérer les documents associés.
4. **Jugement de pertinence :**
   Les utilisateurs évaluent manuellement les documents récupérés. Chaque document est classé comme pertinent (1) ou non pertinent (0), selon la perception de l'utilisateur.
5. **Évaluation :**
   L'évaluation repose sur le calcul de la précision (*precision*) sur les 5, 10 et 15 premiers documents retournés, afin de mesurer la qualité des résultats en tête de classement. La métrique de précision à un seuil k ($P@k$) est mathématiquement définie par :

$$P@k = \frac{\text{Nombre de documents pertinents parmi les } k \text{ premiers documents}}{k} \tag{III.1}$$

Signification des paramètres

Le calcul de la métrique repose sur les paramètres clés suivants :

- $P@k$ **(*Precision at k*)** : Représente la précision calculée sur les k premiers résultats retournés par le système.
- k : Correspond au nombre de documents pris en compte dans le classement (par exemple 5, 10 ou 15).
- **Nombre de documents pertinents** : Désigne les documents jugés pertinents par l'utilisateur parmi les k premiers résultats affichés.

6. Comparaison :

Nous comparons les performances selon deux scénarios principaux :

(A) recherche d'information classique basée sur la requête initiale.

(B) recherche d'information personnalisée (RIP-LLM) basée sur l'exploitation du profil utilisateur et la reformulation des requêtes.

Cette comparaison permet d'évaluer l'impact de la personnalisation sur la qualité des résultats de recherche.

6.3 Expérimentation

L'expérimentation consiste à exécuter les requêtes des utilisateurs sur notre modèle de recherche selon les scénarios définis dans la stratégie d'évaluation. L'objectif principal est d'analyser l'impact de l'exploitation du profil utilisateur sur l'amélioration de la pertinence des résultats retournés par le système.

Construction des profils et requêtes

Le tableau (**Tableau 3.1**) présente les utilisateurs, leurs centres d'intérêt, leur localisation, les termes du profil ainsi que les requêtes initiales utilisées dans l'expérimentation.

Chaque utilisateur est représenté par un profil contenant des informations contextuelles, des préférences et des centres d'intérêt.

À partir de ces profils, des termes représentatifs sont extraits afin de caractériser le contenu informationnel de chaque utilisateur.

Ces termes sont ensuite utilisés pour construire une représentation pondérée du profil.

Enfin, chaque utilisateur dispose de requêtes initiales correspondant à ses besoins informationnels, utilisant des termes et des mots-clés issus du profil afin d'améliorer la pertinence des résultats de recherche.

Table III.1: Profils utilisateurs, termes extraits et requêtes reformulées (RIP-LLM)

Utilisateur	**Centres d'intérêt**	**Loc.**	**Termes du profil**	**Détails (Termes + Reformulations)**	**Requêtes initiales**
U1 (id_user = 1)	la recherche d'information	Médéa	`recherche` `information` `document` `tfidf` `indexation`	**Termes (TF-IDF) :** recherche : 0.521, information : 0.500, document : 0.108, tfidf : 0.751, indexation : 0.654 **Requêtes reformulées :** – système de recherche d'information indexation – calcul de pondération TF-IDF algorithme	informatique Python SiliconValley
U2 (id_user = 2)	patronage	Médéa	`patronage` `couture` `tissu` `confection` `mesure`	**Termes (TF-IDF) :** patronage : 0.680, couture : 0.550, tissu : 0.420, confection : 0.390, mesure : 0.310 **Requêtes reformulées :** – technique de patronage et coupe couture – guide de confection de vêtements sur mesure	patronage coupe et couture
U3 (id_user = 3)	médecine	Blida	`medicine` `anatomy` `medical` `microbiology`	**Termes (TF-IDF) :** medicine : 0.720, anatomy : 0.650, medical : 0.580, microbiology : 0.440, patient : 0.410 **Requêtes reformulées :** – cours anatomie et physiologie humaine – manuel de microbiologie médicale pdf	anatomie et physiologie medical anatomy 1
U4 (id_user = 4)	le ballon, les matchs	Alger	`intelligence_artificielle` `deep_learning` `ia_generative`	**Termes (TF-IDF) :** intelligence_artificielle : 0.750, deep_learning : 0.680, ia_generative : 0.610, reseaux_neuronaux : 0.320 **Requêtes reformulées :** – introduction à l'intelligence artificielle générative – architecture des réseaux de neurones profonds	intelligence artificielle reseaux neuronaux
U5 (id_user = 5)	la plomberie, travaille	Oran	`html` `css` `javascript` `frontend`	**Termes (TF-IDF) :** html : 0.620, css : 0.580, javascript : 0.550, frontend : 0.490, application_web : 0.280 **Requêtes reformulées :** – tutoriel développement web frontend HTML CSS – conception d'applications web modernes en JavaScript	html css js frontend
U6 (id_user = 6)	la nature, durable	Constantine	`tourisme` `tourisme_durable` `destination`	**Termes (TF-IDF) :** tourisme : 0.690, tourisme_durable : 0.640, destination : 0.510, culturel : 0.270 **Requêtes reformulées :** – stratégies de marketing pour tourisme durable – gestion des destinations touristiques et culturelles	*Vide* tourisme durable
U7 (id_user = 7)	la cuisine, la pâtisserie	Tlemcen	`biodiversite` `nature` `ecologie` `ecosystemes`	**Termes (TF-IDF) :** biodiversite : 0.710, nature : 0.660, ecologie : 0.590, ecosystemes : 0.520, conservation : 0.310 **Requêtes reformulées :** – protection de la biodiversité et des écosystèmes – solutions fondées sur la nature et climat	biodiversite et nature solutions fondees

Continued on next page

Table III.1: Profils utilisateurs, termes extraits et requêtes reformulées (RIP-LLM) (Continued)

Utilisateur	**Centres d'intérêt**	**Loc.**	**Termes du profil**	**Détails (Termes + Reformulations)**	**Requêtes initiales**
U8 (id_user = 8)	développement web	Blida	`biology` `cellule` `genetique` `concepts_biologie`	**Termes (TF-IDF) :** biology : 0.680, cellule : 0.610, genetique : 0.570, concepts_biologie : 0.280 **Requêtes reformulées :** – introduction à la biologie cellulaire openstax – recherche en génétique des organismes vivants	introduction biology pdf biologie cellulaire
U9 (id_user = 9)	deep learning, ia	Sétif	`machine_learning` `deep_learning` `algorithmes`	**Termes (TF-IDF) :** machine_learning : 0.740, deep_learning : 0.710, algorithmes : 0.590, apprentissage_supervise : 0.340 **Requêtes reformulées :** – algorithmes d'apprentissage supervisé et classification – modèles de prédiction en machine learning	machine learning pdf classification ML
U10 (id_user = 10)	data science, python	Annaba	`data_science` `analyse_donnees` `visualisation`	**Termes (TF-IDF) :** data_science : 0.780, analyse_donnees : 0.690, visualisation : 0.550, big_data : 0.330 **Requêtes reformulées :** – techniques d'analyse de données et statistiques – outils de visualisation en data science Python	introduction data science pdf analyse de donnees

6.3.1 Résultats et discussion

Dans cette partie, nous présentons les résultats expérimentaux obtenus.

Le tableau suivant (Tableau 3.2) illustre la précision obtenue aux rangs 5, 10 et 15 documents retournés, pour chaque requête, chaque utilisateur et chaque scénario étudié.

Nous comparons les performances du système selon deux scénarios principaux :

(A) Recherche d'information classique : Évaluation basée uniquement sur la requête initiale formulée par l'utilisateur, sans aucune adaptation contextuelle.

(B) Recherche d'information personnalisée (RIP-LLM) : Évaluation basée sur l'exploitation dynamique du profil utilisateur et la reformulation contextuelle des requêtes via le modèle de langage.

Cette comparaison permet d'évaluer de manière quantitative l'impact de la personnalisation sur la qualité et la pertinence des résultats de recherche.

Table III.2: Comparaison des performances (P@5, P@10, P@15) entre la recherche classique (A) et personnalisée RIP-LLM (B)

Utilisateurs / Requêtes Initiales	**(A) Recherche Classique**			**(B) Recherche Personnalisée (RIP-LLM)**		
	P@5	**P@10**	**P@15**	**P@5**	**P@10**	**P@15**
User 1						
informatique	0,2	0,1	0,66	0,8	0,8	0,866
python	0	0,1	0,066	1	0,9	0,866
siliconvalley	0	0,1	0,066	1	0,9	0,866
Moyen Précision	**0,066**	**0,1**	**0,264**	**0,933**	**0,867**	**0,866**
User 2						
patronage	0	0	0	1	0,9	0,866
coupe et couture	0	0,1	0,066	0,8	0,7	0,666
Moyen Précision	**0**	**0,05**	**0,033**	**0,9**	**0,8**	**0,766**
User 3						
anatomie et physiologie	0	0,3	0,266	1	1	0,866
medical anatomy 1	0	0,1	0,066	0,4	0,5	0,533
Moyen Précision	**0**	**0,2**	**0,166**	**0,7**	**0,75**	**0,699**
User 4						
intelligence artificielle	0,2	0,4	0,4	1	1	1
reseaux neuronaux	0,4	0,3	0,2	0,8	0,6	0,466
Moyen Précision	**0,3**	**0,35**	**0,3**	**0,9**	**0,8**	**0,733**
User 5						
html css js	0	0	0	1	1	1
frontend	0	0	0	1	1	0,866

Continued on next page

Table III.2: Comparaison des performances (P@5, P@10, P@15) entre la recherche classique (A) et personnalisée RIP-LLM (B) (Continued)

Utilisateurs / Requêtes Initiales	**(A) Recherche Classique**			**(B) Recherche Personnalisée (RIP-LLM)**		
	P@5	**P@10**	**P@15**	**P@5**	**P@10**	**P@15**
Moyen Précision	**0**	**0**	**0**	**1**	**1**	**0,933**
User 6						
nature	0	0.1	0.066	1	0,9	0,6
tourisme durable	0	0	0,1	0,8	0,7	0,666
Moyen Précision	**0**	**0,05**	**0,083**	**0,9**	**0,8**	**0,633**
User 7						
biodiversite et nature	0,2	0,1	0,066	1	0,9	0,866
solutions fondees	0	0	0	0,8	0,8	0,8
Moyen Précision	**0,1**	**0,05**	**0,033**	**0,9**	**0,85**	**0,833**
User 8						
introduction biology pdf	0	0,2	0,133	0,8	0,8	0,866
biologie cellulaire	0	0	0	0,6	0,7	0,666
Moyen Précision	**0**	**0,1**	**0,066**	**0,7**	**0,75**	**0,766**
User 9						
machine learning pdf	0	0	0	1	0,9	0,933
classification ML	0,2	0,1	0,066	0,8	0,8	0,8
Moyen Précision	**0,1**	**0,05**	**0,033**	**0,9**	**0,85**	**0,866**
User 10						
introduction data science pdf	0	0	0	1	1	0,933
analyse de donnees	0	0,2	0,133	1	0,9	0,866
Moyen Précision	**0**	**0,1**	**0,066**	**1**	**0,95**	**0,9**

6.4 Exemples de processus de reformulation

Voici quelques exemples (Tableau III.3) illustrant le processus de reformulation des requêtes dans notre système, en comparant la requête initiale (Q) du scénario (A) avec sa version reformulée et personnalisée du scénario (B), basée sur le profil utilisateur (centres d'intérêt et termes extraits).

Table III.3: Exemples de processus de reformulation personnalisée (RIP-LLM) selon les deux scénarios

Utilisateur	(A) Requête initiale (Q)	Termes du profil	(B) Requête reformulée (RIP-LLM)
U1	informatique	AI, python, data, informatique	Q' = Domaines d'étude en informatique et recherche d'information
			Q' = Systèmes et méthodes informatiques de récupération d'information
U6	nature	tourisme, tourisme_durable, destination	Q' = Documentation pdf sur le tourisme durable et la préservation de la nature
			Q' = Guides pratiques pour la protection des forêts et des écosystèmes naturels
U3	anatomie, physiologie	médecine, anatomie, santé	Q' = cours d'anatomie et physiologie humaine
			Q' = introduction à la biologie médicale et structures du corps humain
...	...	...	...
			...

Nous remarquons que cette illustration met en évidence le processus de reformulation des requêtes basé sur l'exploitation du profil utilisateur dans notre approche de recherche d'information personnalisée (RIP-LLM). La reformulation permet d'enrichir la requête initiale en intégrant les termes extraits du profil, tels que les centres d'intérêt et les préférences de

l'utilisateur, afin d'améliorer la pertinence des résultats retournés.

Dans notre système, la reformulation repose sur l'adaptation de la requête initiale (Q) en une requête enrichie (Q'), en utilisant les termes du profil utilisateur et la localisation. Par exemple, pour un utilisateur ayant un intérêt en intelligence artificielle et en data science, la requête initiale simple est transformée en une requête plus spécifique intégrant ces concepts afin d'orienter la recherche vers des résultats plus pertinents.

Cette reformulation personnalisée permet ainsi d'améliorer la correspondance entre les besoins informationnels de l'utilisateur et les documents retournés par le système.

Ci-dessous, nous présentons quelques exemples illustrant les améliorations des résultats obtenus après la reformulation des requêtes dans notre approche de personnalisation.

Exemple 01 :

Les premiers résultats de la requête « nature» envoyée sans reformulation par l'utilisateur 6,dont le profil est centré sur le tourisme durable, ne satisfont pas ses besoins réels (Figure 3.6). À titre d'exemple, les documents d'organisations financières ou de guides d'édition de revues scientifiques ne présentent aucune pertinence pour ses recherches axées sur l'écotourisme.

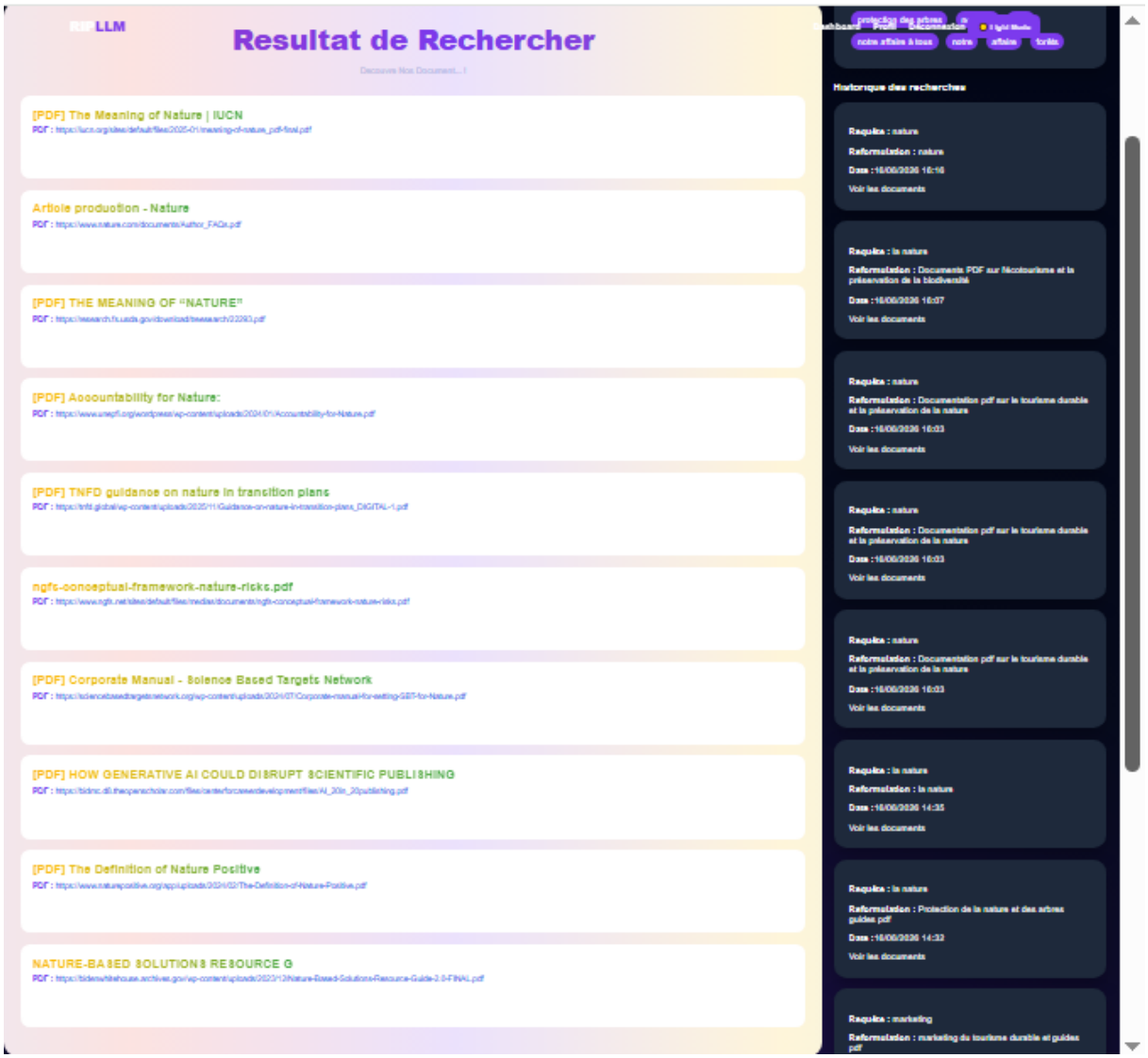


FIGURE III.16 – Représentation des résultats bruts (Scénario A) pour la requête « nature » de l'utilisateur 6.

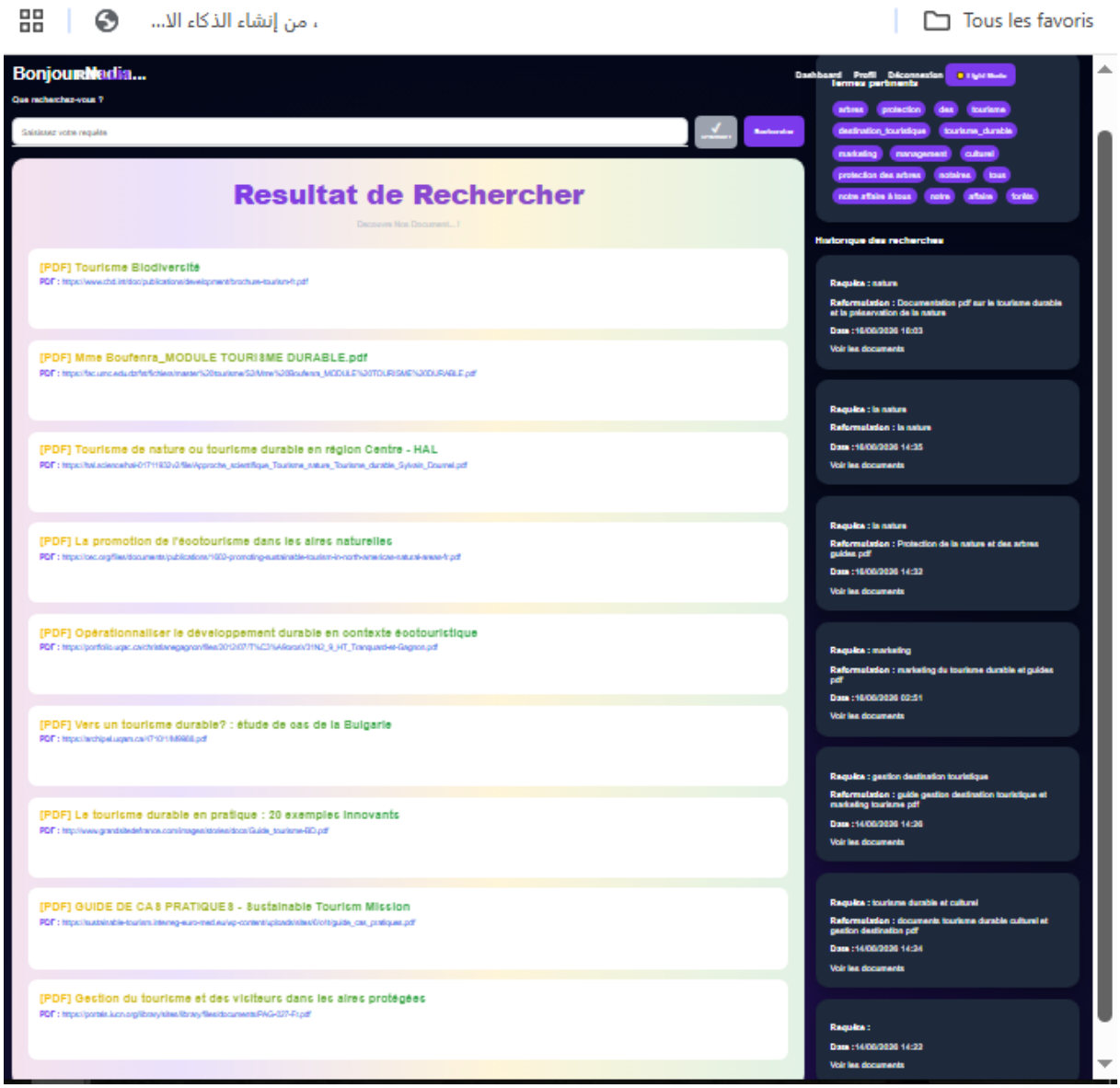


FIGURE III.17 – Résultats personnalisés obtenus via l'approche RIP-LLM pour l'utilisateur 6 après reformulation.

Nous constatons que, suite à cette reformulation personnalisée de la requête, la pertinence des résultats s'est considérablement améliorée, répondant ainsi de manière très satisfaisante aux besoins spécifiques de l'utilisateur.

6.5 Résultats globaux de précision

Pour conclure notre phase d'évaluation, nous présentons dans le tableau suivant (**Tableau 3.4**) une synthèse globale des performances de notre système. Ce tableau affiche la moyenne des précisions ($P@5$, $P@10$ et $P@15$) calculée sur l'ensemble des requêtes soumises par tous les utilisateurs, permettant ainsi de comparer l'efficacité des deux scénarios expérimentés.
L'analyse de ce tableau montre une progression remarquable des performances de recherche. En effet, la précision moyenne générale (P_{moy}) passe de **0.179** dans le scénario classique (A) à **0.523** dans le scénario personnalisé (B).

TABLE III.4 – Synthèse globale des précisions moyennes ($P@K$) selon les deux scénarios.

Métriques de Précision	**Scénario (A)** *Sans reformulation (Classique)*	**Scénario (B)** *Avec approche RIP-LLM*
P@5	0.173	0.553
P@10	0.190	0.506
P@15	0.175	0.509
P_{moy}	**0.179**	**0.523**

Ces résultats quantitatifs confirment clairement l'efficacité de l'approche **RIP-LLM**. L'implication interactive de l'utilisateur dans le choix de la reformulation, optimisée par la puissance sémantique des grands modèles de langage (LLM), permet d'affiner considérablement l'espace de recherche et de lever efficacement les ambiguïtés des requêtes initiales par rapport au modèle traditionnel.

6.5.1 Structure de la base de données

Les différentes composantes du profil utilisateur ainsi que l'historique des requêtes sont implémentées et stockées physiquement dans notre base de données selon la structure illustrée par la **Figure III.18** :

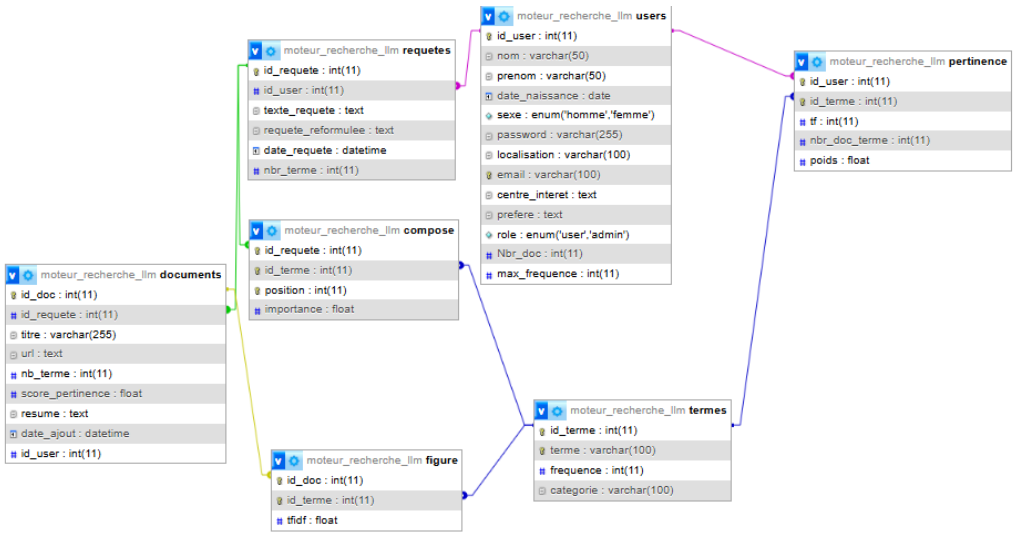


FIGURE III.18 – la base de données

a) Les entités :

N°	Entité	Identifiant	Propriétés et types
1	**Users**	Id_user	- Nom (VARCHAR), Prenom (VARCHAR), Date_naissance (DATE) - Sexe (ENUM), Password (VARCHAR), Localisation (VARCHAR) - Email (VARCHAR), Centre_interet (TEXT), Prefere (TEXT), Role (ENUM)
2	**Termes**	Id_terme	- Terme (VARCHAR), Frequence (INT), Categorie (VARCHAR)
3	**Requetes**	Id_requete	- Texte_requete (TEXT), Requete_reformulee (TEXT) - Date_requete (DATETIME), Nbr_terme (INT)
4	**Documents**	Id_doc	- Titre (VARCHAR), Url (TEXT), Nb_terme (INT) - Score_pertinence (FLOAT), Resume (TEXT), Date_ajout (DATETIME)

b) Les relations :

N°	Relation	Entités participants	Identifiant	Propriétés
1	**Compose**	Requetes, Termes	Id_requete, Id_terme	- Position (INT), -Importance (FLOAT)
2	**Figure**	Documents, Termes	Id_doc, Id_terme	- Tfidf (FLOAT)
3	**Pertinence**	Users, Requetes, Documents	Id_user, Id_requete, Id_doc	- Est_pertinent (BOOLEAN/INT)
4	**Requetes**	Users	Id_requete	*(Id_user : clé étrangère)*
5	**Documents**	Requetes	Id_doc	*(Id_requete*, *Id_user : clés étrangères)*

6.5.2 Création des tables de la base de données

Notre base de données relationnelle est implémentée sous MySQL et comprend six tables principales permettant la gestion des utilisateurs, des requêtes, des termes ainsi que des documents récupérés par le moteur de recherche personnalisé.

La Figure III.19 présente l'ensemble des tables constituant la base de données.

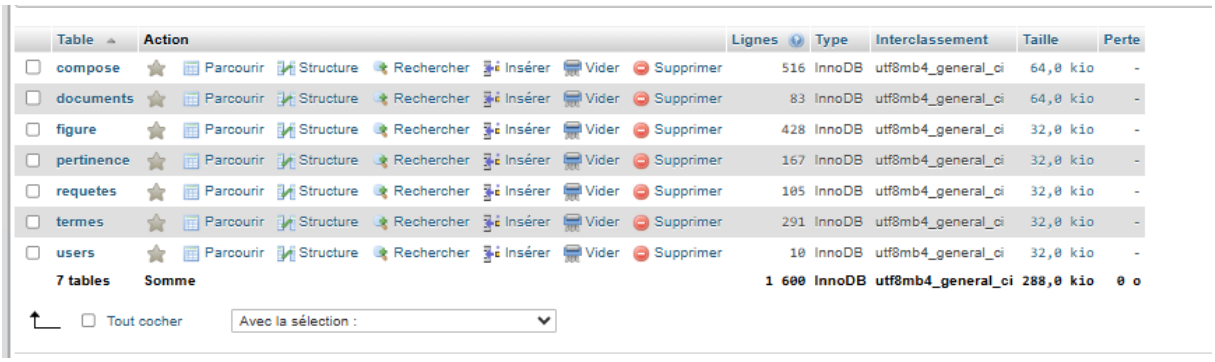


Table	Action	Lignes	Type	Interclassement	Taille	Perte
compose	Parcourir Structure Rechercher Insérer Vider Supprimer	516	InnoDB	utf8mb4_general_ci	64,0 kio	-
documents	Parcourir Structure Rechercher Insérer Vider Supprimer	83	InnoDB	utf8mb4_general_ci	64,0 kio	-
figure	Parcourir Structure Rechercher Insérer Vider Supprimer	428	InnoDB	utf8mb4_general_ci	32,0 kio	-
pertinence	Parcourir Structure Rechercher Insérer Vider Supprimer	167	InnoDB	utf8mb4_general_ci	32,0 kio	-
requetes	Parcourir Structure Rechercher Insérer Vider Supprimer	105	InnoDB	utf8mb4_general_ci	32,0 kio	-
termes	Parcourir Structure Rechercher Insérer Vider Supprimer	291	InnoDB	utf8mb4_general_ci	32,0 kio	-
users	Parcourir Structure Rechercher Insérer Vider Supprimer	10	InnoDB	utf8mb4_general_ci	32,0 kio	-
7 tables	Somme	1 600	InnoDB	utf8mb4_general_ci	288,0 kio	0 o

Tout cocher — Avec la sélection :

[Architecture globale du schéma relationnel (Tables et clés)]

FIGURE III.19 – Les tables de la base de données.

6.6 Liste des tables de la base de données

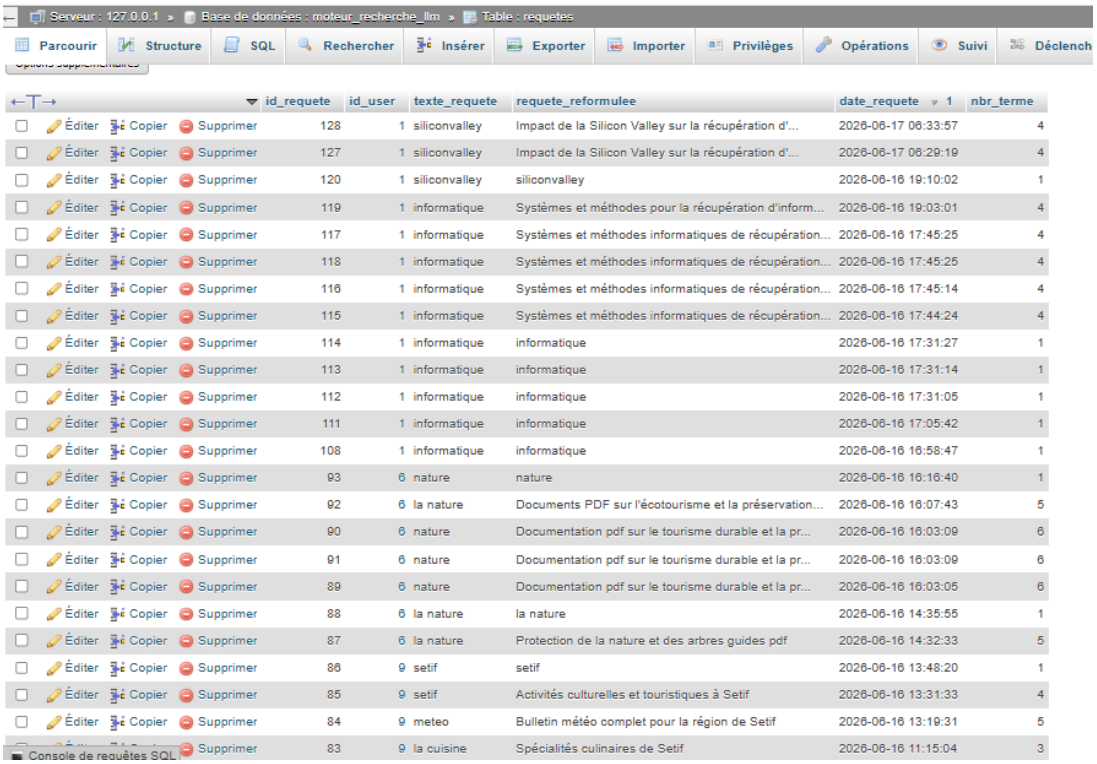
Notre base de données comprend les tables suivantes :

• **Table « Users »** Cette table contient les informations relatives aux utilisateurs du système. Elle stocke les données personnelles nécessaires à l'identification et à la personnalisation de la recherche, telles que le nom, le prénom, l'adresse électronique, la localisation ainsi que les centres d'intérêt de l'utilisateur.

	id_user	nom	prenom	date_naissance	sexe	password	localisation	email	centre_interet	prefere	role	Nbr_doc	max_frequence
Edit Copy Delete	1	Meriem	Guesmia	2002-11-27	femme	scrypt:32768:8:1$K8Id5c9Id0BqUGgd$86c8f7d432d9b152...	medea	meriemeriem27@gmail.com	la recharche d'information		user	10	168
Edit Copy Delete	2	safa	fellague	2003-11-14	femme	scrypt:32768:8:1$BdzeMkl92MpfuL9c$6e179c2b09bcff4b...	medea	safafellague@gmail.com	les accessoires , les bijoux		user	5	10
Edit Copy Delete	3	khadidja	fellague	2006-11-07	femme	scrypt:32768:8:1$UPDRMje6ygXpG8lp$15b129ecfac2786a...	blida	khadidjafellague@gmail.com	medcine		user	6	9
Edit Copy Delete	4	Amine	Benali	2001-05-12	homme	scrypt:32768:8:1$GxbMsNVUAUQKPe6I$8f001643872eadd1...	Alger	amine.benali@gamil.com	le ballon ,les matchs	documents pdf, cours universitaires	user	12	6
Edit Copy Delete	5	Yacine	Bouzid	2000-07-21	homme	scrypt:32768:8:1$I5xIM9IAvPOe12ZT$e7c772ebd6f54625...	Oran	yacine.bouzid@gmail.com	la plomberies ,travaille	tutoriels pratiques, exemples	user	6	6
Edit Copy Delete	6	Nadia	Hamdi	2002-03-10	femme	scrypt:32768:8:1$3lK0TWZHO4kWZZXP$0cd3347983729825...	Constantine	nadia.hamdi@gmail.com	la nature ,protege les arbres	guides pdf, tourisme durable	user	8	7
Edit Copy Delete	7	Sabrina	Kaci	2001-08-18	femme	scrypt:32768:8:1$pNFcT6zktQYw8jdK$fea961fc23bff659...	Tlemcen	sabrina.kaci@gmail.com	la cuisine ,la pâtisserie	biodiversite, environnement	user	6	7
Edit Copy Delete	8	Karim	Mansouri	2000-11-11	homme	scrypt:32768:8:1$MSZ40NGXiOt5V8DI$114a0d9e0b7121a6...	Blida	karim.mansouri@gmail.com	devlopement web	cours biologie, cellules	user	6	6
Edit Copy Delete	9	Lina	Meziane	2003-02-25	femme	scrypt:32768:8:1$5RyCS3o2YovOzX0z$a4f19213e12efd54...	Setif	lina.meziane@gmail.com	deep learning ,intelligence artificiel	algorithmes, classification	user	13	43
Edit Copy Delete	10	Riad	Cherif	2001-09-14		scrypt:32768:8:1$gWhVpUrYrGjfrf2l$90883dbd0c7abe8d...	Annaba	riad.cherif@test.com	data science ,python ,java	analyse de donnees, visualisation	user	10	43

FIGURE III.20 – Exemple de la table « Users ».

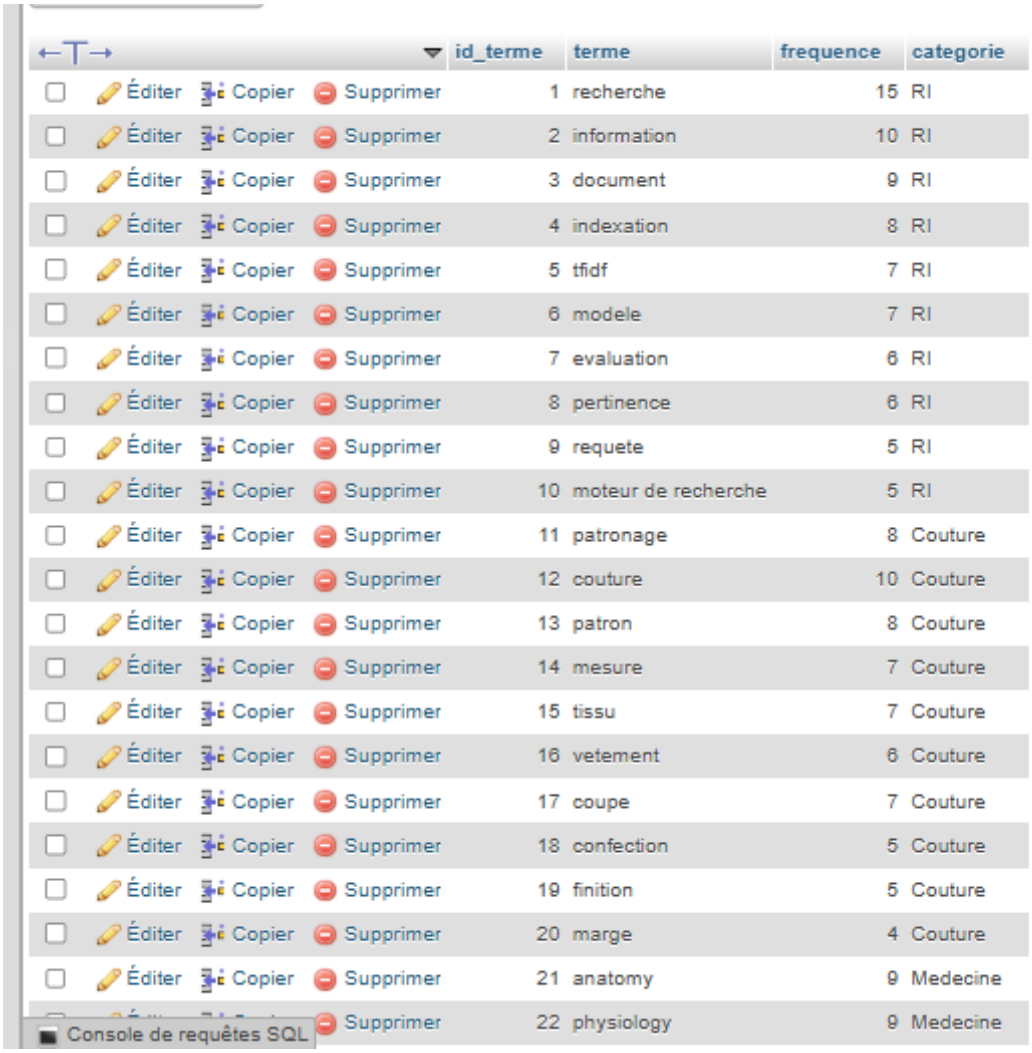
Table « Requetes » Cette table enregistre les requêtes soumises par les utilisateurs ainsi que leurs versions reformulées générées par le système. Elle conserve également la date de la requête et le nombre de termes qui la composent.



	id_requete	id_user	texte_requete	requete_reformulee	date_requete	nbr_terme
Éditer Copier Supprimer	128	1	siliconvalley	Impact de la Silicon Valley sur la récupération d'...	2026-06-17 06:33:57	4
Éditer Copier Supprimer	127	1	siliconvalley	Impact de la Silicon Valley sur la récupération d'...	2026-06-17 06:29:19	4
Éditer Copier Supprimer	120	1	siliconvalley	siliconvalley	2026-06-16 19:10:02	1
Éditer Copier Supprimer	119	1	informatique	Systèmes et méthodes pour la récupération d'inform...	2026-06-16 19:03:01	4
Éditer Copier Supprimer	117	1	informatique	Systèmes et méthodes informatiques de récupération...	2026-06-16 17:45:25	4
Éditer Copier Supprimer	118	1	informatique	Systèmes et méthodes informatiques de récupération...	2026-06-16 17:45:25	4
Éditer Copier Supprimer	116	1	informatique	Systèmes et méthodes informatiques de récupération...	2026-06-16 17:45:14	4
Éditer Copier Supprimer	115	1	informatique	Systèmes et méthodes informatiques de récupération...	2026-06-16 17:44:24	4
Éditer Copier Supprimer	114	1	informatique	informatique	2026-06-16 17:31:27	1
Éditer Copier Supprimer	113	1	informatique	informatique	2026-06-16 17:31:14	1
Éditer Copier Supprimer	112	1	informatique	informatique	2026-06-16 17:31:05	1
Éditer Copier Supprimer	111	1	informatique	informatique	2026-06-16 17:05:42	1
Éditer Copier Supprimer	108	1	informatique	informatique	2026-06-16 16:58:47	1
Éditer Copier Supprimer	93	6	nature	nature	2026-06-16 16:16:40	1
Éditer Copier Supprimer	92	6	la nature	Documents PDF sur l'écotourisme et la préservation...	2026-06-16 16:07:43	5
Éditer Copier Supprimer	90	6	nature	Documentation pdf sur le tourisme durable et la pr...	2026-06-16 16:03:09	6
Éditer Copier Supprimer	91	6	nature	Documentation pdf sur le tourisme durable et la pr...	2026-06-16 16:03:09	6
Éditer Copier Supprimer	89	6	nature	Documentation pdf sur le tourisme durable et la pr...	2026-06-16 16:03:05	6
Éditer Copier Supprimer	88	6	la nature	la nature	2026-06-16 14:35:55	1
Éditer Copier Supprimer	87	6	la nature	Protection de la nature et des arbres guides pdf	2026-06-16 14:32:33	5
Éditer Copier Supprimer	86	9	setif	setif	2026-06-16 13:48:20	1
Éditer Copier Supprimer	85	9	setif	Activités culturelles et touristiques à Setif	2026-06-16 13:31:33	4
Éditer Copier Supprimer	84	9	meteo	Bulletin météo complet pour la région de Setif	2026-06-16 13:19:31	5
Supprimer	83	9	la cuisine	Spécialités culinaires de Setif	2026-06-16 11:15:04	3

FIGURE III.21 – Exemple de la table « Requetes ».

Table « Termes » Cette table contient l'ensemble des termes indexés dans le système. Pour chaque terme, sont enregistrés sa fréquence d'apparition ainsi que sa catégorie sémantique.



	id_terme	terme	frequence	categorie
Éditer Copier Supprimer	1	recherche	15	RI
Éditer Copier Supprimer	2	information	10	RI
Éditer Copier Supprimer	3	document	9	RI
Éditer Copier Supprimer	4	indexation	8	RI
Éditer Copier Supprimer	5	tfidf	7	RI
Éditer Copier Supprimer	6	modele	7	RI
Éditer Copier Supprimer	7	evaluation	6	RI
Éditer Copier Supprimer	8	pertinence	6	RI
Éditer Copier Supprimer	9	requete	5	RI
Éditer Copier Supprimer	10	moteur de recherche	5	RI
Éditer Copier Supprimer	11	patronage	8	Couture
Éditer Copier Supprimer	12	couture	10	Couture
Éditer Copier Supprimer	13	patron	8	Couture
Éditer Copier Supprimer	14	mesure	7	Couture
Éditer Copier Supprimer	15	tissu	7	Couture
Éditer Copier Supprimer	16	vetement	6	Couture
Éditer Copier Supprimer	17	coupe	7	Couture
Éditer Copier Supprimer	18	confection	5	Couture
Éditer Copier Supprimer	19	finition	5	Couture
Éditer Copier Supprimer	20	marge	4	Couture
Éditer Copier Supprimer	21	anatomy	9	Medecine
Supprimer	22	physiology	9	Medecine

Console de requêtes SQL

FIGURE III.22 – Exemple de la table « Termes ».

• **Table « Documents »** Cette table stocke les documents récupérés suite à l'exécution des requêtes. Chaque document est caractérisé par son titre, son URL, son résumé, son score de pertinence ainsi que sa date d'ajout.



	id_doc	id_requete	titre	url	nb_terme	score_pertinence	resume	date_ajout	id_user
Éditer Copier Supprimer	1	1	Introduction to Information Retrieval - Stanford	https://nlp.stanford.edu/IR-book/pdf/irbookonliner...	150	0.98	Livre PDF complet sur l'indexation, TF-IDF, modèle...	2026-06-02 11:46:51	1
Éditer Copier Supprimer	2	1	Introduction à la Recherche d'Information	https://fr.scribd.com/document/979330542/II-Inform...	90	0.95	Introduction aux systèmes de recherche d'informati...	2026-06-02 11:46:51	1
Éditer Copier Supprimer	3	2	Recherche d'information - Université de Biskra	https://elearning.univ-biskra.dz/moodle2024/mod/re...	80	0.94	Cours universitaire sur les mots-clés, les requête...	2026-06-02 11:46:51	1
Éditer Copier Supprimer	4	3	Information Retrieval Systems	https://mist.ac.in/MIST%20ACADEMIC/files/CSM/3%20Y...	130	0.96	Cours PDF sur les systèmes de recherche d'informat...	2026-06-02 11:46:51	1
Éditer Copier Supprimer	5	2	Méthodologie de recherche d'information	https://coop-ist.cirad.fr/content/download/4205/31...	70	0.91	Document PDF sur la méthodologie de recherche docu...	2026-06-02 11:46:51	1
Éditer Copier Supprimer	6	4	Cours Complet de Patronage Couture	https://fr.scribd.com/document/938444473/COURS-DE-...	140	0.97	Cours sur les fondamentaux du patronage : prise de...	2026-06-03 10:20:00	2
Éditer Copier Supprimer	7	5	Manuel de Coupe et Couture : Guide pour la 1ère An...	https://manuels-scolaires-rdc.com/la-coupe-1-ere-a...	125	0.94	Manuel pour apprendre la prise de mesures corporel...	2026-06-03 10:21:00	2
Éditer Copier Supprimer	8	6	Documents PDF gratuits - Coupe et Couture Loisirs	https://www.coupe-et-couture-loisirs.fr/ressources...	130	0.91	Page contenant des documents PDF téléchargeables e...	2026-06-03 10:22:00	2
Éditer Copier Supprimer	9	7	Lexique de la Couture PDF	https://fr.scribd.com/document/945258376/0-Lexique...	165	0.88	Lexique détaillé des termes utilisés dans la coutu...	2026-06-03 10:23:00	2
Éditer Copier Supprimer	10	8	Lexique couture : vocabulaire à connaître	https://leslubiesdelouise.com/2024/11/25/vocabulai...	150	0.86	Vocabulaire de couture avec des termes comme coutu...	2026-06-03 10:24:00	2
Éditer Copier Supprimer	11	9	Anatomy and Physiology - OpenStax	https://d3bxy9euw4e147.cloudfront.net/oscms-ga/med...	320	0.99	Official core textbook of Anatomy and Physiology b...	2026-06-03 11:10:00	3

Console de requêtes SQL

FIGURE III.23 – Exemple de la table « Documents ».

• **Table « Compose »** Cette table représente la relation entre les requêtes et les termes qui les composent. Elle permet de conserver la position de chaque terme dans la requête ainsi que son

degré d'importance.

	id_requete	1	id_terme	position	importance
Éditer Copier Supprimer	128		460	3	0.333333
Éditer Copier Supprimer	128		459	2	0.5
Éditer Copier Supprimer	128		458	1	1
Éditer Copier Supprimer	128		456	4	0.25
Éditer Copier Supprimer	127		460	3	0.333333
Éditer Copier Supprimer	127		459	2	0.5
Éditer Copier Supprimer	127		458	1	1
Éditer Copier Supprimer	127		456	4	0.25
Éditer Copier Supprimer	120		457	1	1
Éditer Copier Supprimer	119		456	3	0.333333
Éditer Copier Supprimer	119		454	1	1
Éditer Copier Supprimer	119		453	4	0.25
Éditer Copier Supprimer	119		291	2	0.5
Éditer Copier Supprimer	118		456	4	0.25
Éditer Copier Supprimer	118		455	3	0.333333
Éditer Copier Supprimer	118		454	1	1
Éditer Copier Supprimer	118		291	2	0.5
Console de requêtes SQL Supprimer	117		456	4	0.25

FIGURE III.24 – Exemple de la table « Compose ».

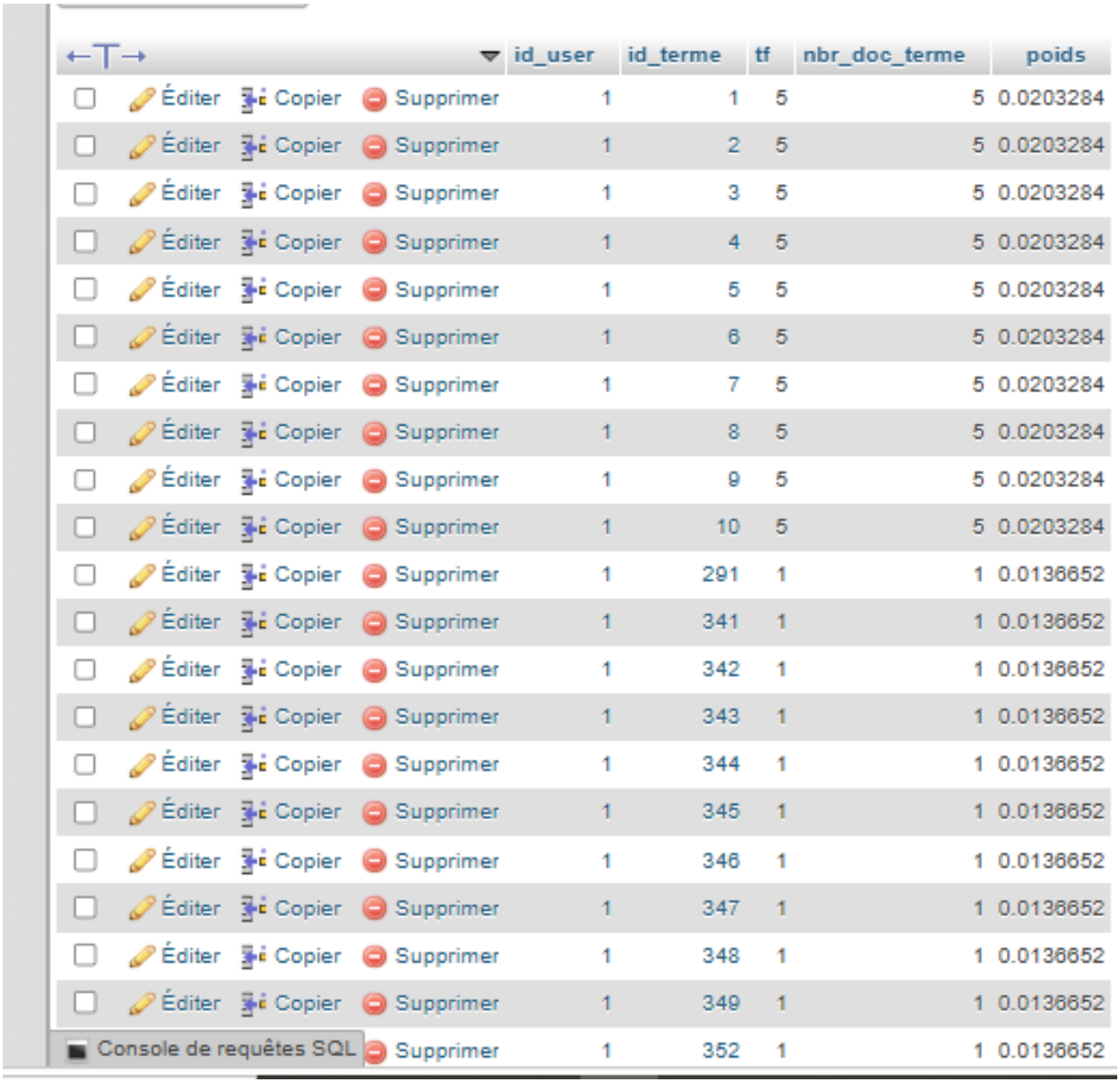
• **Table « Figure »** Cette table établit la relation entre les documents et les termes qu'ils contiennent. Elle stocke pour chaque couple document-terme le poids associé ainsi que la valeur TF-IDF utilisée dans le processus d'indexation et de calcul de pertinence.



	id_doc	id_terme	tfidf
Éditer Copier Supprimer	1	1	0.95
Éditer Copier Supprimer	1	2	0.94
Éditer Copier Supprimer	1	3	0.9
Éditer Copier Supprimer	1	4	0.88
Éditer Copier Supprimer	1	5	0.86
Éditer Copier Supprimer	1	6	0.84
Éditer Copier Supprimer	1	7	0.82
Éditer Copier Supprimer	1	8	0.8
Éditer Copier Supprimer	1	9	0.78
Éditer Copier Supprimer	1	10	0.76
Éditer Copier Supprimer	2	1	0.93
Éditer Copier Supprimer	2	2	0.91
Éditer Copier Supprimer	2	3	0.88
Éditer Copier Supprimer	2	4	0.85
Éditer Copier Supprimer	2	5	0.73
Éditer Copier Supprimer	2	6	0.82

FIGURE III.25 – Exemple de la table « Figure ».

• **Table « Pertinence »** Cette table joue un rôle crucial dans la phase d'évaluation du système (calcul des métriques de précision P@K, MAP, nDCG). Elle stocke les jugements de pertinence explicites ou implicites associant un utilisateur, une requête spécifique et un document retourné, indiquant si le document répond précisément au besoin d'information.



	id_user	id_terme	tf	nbr_doc_terme	poids
Éditer Copier Supprimer	1	1	5	5	0.0203284
Éditer Copier Supprimer	1	2	5	5	0.0203284
Éditer Copier Supprimer	1	3	5	5	0.0203284
Éditer Copier Supprimer	1	4	5	5	0.0203284
Éditer Copier Supprimer	1	5	5	5	0.0203284
Éditer Copier Supprimer	1	6	5	5	0.0203284
Éditer Copier Supprimer	1	7	5	5	0.0203284
Éditer Copier Supprimer	1	8	5	5	0.0203284
Éditer Copier Supprimer	1	9	5	5	0.0203284
Éditer Copier Supprimer	1	10	5	5	0.0203284
Éditer Copier Supprimer	1	291	1	1	0.0136652
Éditer Copier Supprimer	1	341	1	1	0.0136652
Éditer Copier Supprimer	1	342	1	1	0.0136652
Éditer Copier Supprimer	1	343	1	1	0.0136652
Éditer Copier Supprimer	1	344	1	1	0.0136652
Éditer Copier Supprimer	1	345	1	1	0.0136652
Éditer Copier Supprimer	1	346	1	1	0.0136652
Éditer Copier Supprimer	1	347	1	1	0.0136652
Éditer Copier Supprimer	1	348	1	1	0.0136652
Éditer Copier Supprimer	1	349	1	1	0.0136652
Console de requêtes SQL Supprimer	1	352	1	1	0.0136652

[Exemple de la table Pertinence]

FIGURE III.26 – Exemple de la table « Pertinence ».

Conclusion

Dans ce chapitre, nous avons présenté la conception et l'implémentation de notre système de recherche d'information personnalisée **RIP-LLM**. Nous avons d'abord décrit la structure du profil utilisateur, composé d'une partie statique et d'une partie dynamique permettant de représenter de manière précise les centres d'intérêt et les besoins informationnels de l'utilisateur.

Nous avons ensuite détaillé le processus de recherche proposé, reposant sur la collaboration de trois agents **LLM** complémentaires :

— L'agent **LLM1** est chargé de préparer le contexte utilisateur à partir de son historique de recherche et des termes les plus représentatifs de son profil.

— L'agent **LLM2** exploite ensuite ce contexte afin de générer des reformulations pertinentes et contextualisées de la requête utilisateur, permettant ainsi d'améliorer la compréhension de ses intentions de recherche.

— Enfin, l'agent **LLM3** intervient après la phase de recherche pour analyser les documents jugés pertinents, en extraire les termes significatifs et mettre à jour dynamiquement le profil utilisateur.

Cette étape garantit l'évolution continue du profil et son adaptation progressive aux centres d'intérêt réels de l'utilisateur.

Par ailleurs, nous avons présenté l'architecture de la base de données mise en place pour assurer la gestion des utilisateurs, des requêtes, des termes et des documents, ainsi que les différentes relations permettant la construction et l'évolution du profil utilisateur. Cette structure garantit une organisation cohérente des données et facilite les opérations de personnalisation et d'évaluation, notamment à travers le calcul des pondérations $TF\text{-}IDF$ et le stockage des jugements de pertinence.

Enfin, les expérimentations réalisées ont démontré l'efficacité de l'approche **RIP-LLM**. Les résultats obtenus montrent une amélioration significative des mesures de précision par rapport à la recherche classique, avec une précision moyenne générale (P_{moy}) passant de 0.179 dans le scénario classique à 0.523 dans le scénario personnalisé. Ces performances confirment l'intérêt de combiner les profils utilisateurs avec les capacités sémantiques des grands modèles de langage afin d'offrir une recherche plus pertinente, plus contextualisée et continuellement adaptée aux besoins réels des utilisateurs.

Chapitre III :

Conclusion générale

Dans ce mémoire, nous avons proposé une approche de recherche d'information personnalisée basée sur l'exploitation des Grands Modèles de Langage (LLMs). L'objectif principal de ce travail est d'améliorer la compréhension du besoin informationnel de l'utilisateur et d'adapter le processus de recherche à son profil ainsi qu'à son contexte, afin de fournir des résultats plus pertinents.

Pour atteindre cet objectif, nous avons conçu le système RIP-LLM, qui repose sur un profil utilisateur combinant des informations générales relativement stables et des informations issues des interactions récentes de l'utilisateur. Cette représentation permet de prendre en compte à la fois les centres d'intérêt à long terme et les besoins ponctuels exprimés au cours des recherches récentes.

Nous avons également proposé un mécanisme d'identification du contexte de recherche permettant de mieux interpréter les requêtes et de réduire leur ambiguïté. À partir de ce contexte et du profil utilisateur, le système génère plusieurs reformulations enrichies de la requête initiale afin de mieux refléter l'intention réelle de l'utilisateur. Celui-ci conserve néanmoins la possibilité de choisir la formulation qui lui paraît la plus adaptée ou de conserver sa requête d'origine.

Afin d'assurer l'évolution continue du système, un mécanisme de mise à jour du profil utilisateur a également été intégré. Celui-ci permet d'enrichir progressivement la représentation des centres d'intérêt de l'utilisateur en tenant compte de ses nouvelles interactions et de l'évolution de ses besoins informationnels.

Les expérimentations réalisées ont montré que l'approche proposée permet de produire des reformulations plus claires, plus explicites et mieux adaptées au contexte de recherche que les requêtes initiales. Les résultats obtenus montrent également que le système conserve des temps de réponse satisfaisants pour une utilisation interactive.

Malgré ces résultats encourageants, plusieurs perspectives peuvent être envisagées. Il serait notamment intéressant d'élargir le cadre d'évaluation à un plus grand nombre d'utilisateurs et à des scénarios de recherche plus variés afin de mieux mesurer les performances du système dans des conditions réelles. L'amélioration de la gestion de l'évolution du profil utilisateur ainsi que l'intégration de mécanismes de retour utilisateur pourraient également contribuer à renforcer la qualité des reformulations générées.

En conclusion, ce travail met en évidence le potentiel des Grands Modèles de Langage dans le domaine de la recherche d'information personnalisée. En combinant le profil utilisateur et le contexte de recherche, le système RIP-LLM contribue à une meilleure compréhension des besoins informationnels et ouvre des perspectives prometteuses pour le développement de systèmes de recherche plus intelligents, plus adaptatifs et davantage centrés sur l'utilisateur.

Limites du travail

Malgré ces résultats encourageants, ce travail présente certaines limites qui ouvrent la voie à des améliorations futures. L'expérimentation a été menée sur un ensemble restreint de profils utilisateurs et de scénarios représentatifs, ce qui ne permet pas de généraliser pleinement les résultats à des contextes d'utilisation plus variés. Par ailleurs, l'évaluation s'est principalement concentrée sur l'impact de la reformulation personnalisée des requêtes à travers les mesures **P@K**, sans intégrer une comparaison expérimentale avec d'autres approches récentes de la littérature, telles que les méthodes de reformulation non basées sur les LLMs, les architectures **RAG** ou les techniques de **reranking neuronal**. Enfin, l'approche proposée repose sur le modèle **Gemini** ; il serait donc intéressant d'étudier son adaptation à des modèles ouverts tels que **Llama 3** ou **Mistral** afin d'analyser le compromis entre les performances, le coût de déploiement et l'autonomie du système.

Perspectives

Les résultats obtenus ouvrent plusieurs perspectives de recherche visant à enrichir davantage l'architecture proposée et à améliorer la personnalisation de la recherche d'information. Une première perspective consiste à faire évoluer le profil utilisateur vers un profil intelligent et dynamique, capable de s'adapter automatiquement aux changements des intérêts de l'utilisateur. Au-delà des informations actuellement exploitées, ce profil pourrait intégrer des tags dynamiques générés automatiquement à partir des requêtes, des documents consultés et des interactions de l'utilisateur. Chaque tag pourrait être associé à un poids reflétant son importance, évoluer au fil du temps et être complété par la détection automatique de nouveaux centres d'intérêt ainsi que par l'identification de relations sémantiques entre les différents domaines explorés.
Une autre perspective consiste à étendre l'architecture proposée par l'intégration d'un nouvel agent intelligent, **AGENT-LLM4**, dédié au post-traitement des résultats de recherche. Cet agent aurait pour mission d'analyser les documents les plus pertinents, d'en produire une synthèse concise et structurée, d'extraire les informations essentielles et de mettre en évidence les éléments les plus utiles en fonction du profil utilisateur. Il pourrait également générer des suggestions de requêtes complémentaires afin d'accompagner l'utilisateur dans l'exploration progressive de son besoin informationnel.
Enfin, il serait intéressant d'évaluer l'approche proposée sur un ensemble plus large de profils utilisateurs, de scénarios de recherche et de collections documentaires, ainsi que d'étudier son adaptation à différents modèles de langage, notamment des modèles open source tels que Llama 3 ou Mistral, afin d'analyser leur impact sur les performances, les coûts de déploiement, les ressources matérielles nécessaires, les possibilités de fine-tuning et la confidentialité des données. Ces perspectives contribueraient à renforcer la robustesse, la flexibilité et l'efficacité des systèmes de recherche d'information personnalisée basés sur les grands modèles de langage.

Références Bibliographiques

[1] Mariam Daoud. *Accès personnalisé à l'information : approche basée sur l'utilisation d'un profil utilisateur sémantique dérivé d'une ontologie de domaines à travers l'historique des sessions de recherche*. PhD thesis, Université Paul Sabatier-Toulouse III, 2009.

[2] Giuseppe Amato and Umberto Straccia. User profile modeling. In *Proceedings of ACM SIGIR*, 1999.

[3] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35 :27730–27744, 2022.

[4] Akari Asai, Zhuosheng Wu, Yizhong Wang, Aviral Sil, Vijay Zhong, and Dragomir Radev. Self-rag : Learning to retrieve, generate, and critique through self-reflection. *International Conference on Learning Representations*, 2024.

[5] Douglas W. Oard and Jinmook Kim. Modeling information content using observable behavior. In *Proceedings of ASIST*, pages 38–45, 2001.

[6] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[7] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 2nd edition, 2011.

[8] Tefko Saracevic. Relevance : A review of the literature (part iii). *JASIST*, 58(13) :2126–2144, 2007.

[9] Jaime Teevan et al. Personalizing search via automated analysis of interests. In *Proceedings of SIGIR*, pages 449–456, 2005.

[10] Peter Brusilovsky and Eva Millán. User models for adaptive hypermedia. In *The Adaptive Web*, pages 3–53. Springer, 2007.

[11] Susan Gauch et al. User profiles for personalized ir. In *The Adaptive Web*, pages 54–89. Springer, 2007.

[12] Stephen E. Robertson and Hugo Zaragoza. The probabilistic relevance framework : Bm25 and beyond. *Foundations and Trends in IR*, 3(4) :333–389, 2009.

[13] Pia Borlund. The concept of relevance in ir. *JASIST*, 54(10) :913–925, 2003.

[14] Peter Ingwersen. Polyrepresentation of information needs. In *Proceedings of SIGIR*, pages 101–110, 1994.

[15] Lobna Hlaoua. *Reformulation de Requêtes par Réinjection de Pertinence dans les Documents Semi-Structurés*. Thèse de doctorat, Université de Toulouse 3, Toulouse, France, dec 2007. Décembre 2007.

[16] Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *CACM*, 18(11) :613–620, 1975.

[17] Stephen E. Robertson and Karen Sparck Jones. Relevance weighting of search terms. *JASIS*, 27(3) :129–146, 1976.

[18] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation. *ACM TOIS*, 20(4) :422–446, 2002.

[19] David Bawden and Lyn Robinson. The dark side of information overload. *Aslib Proceedings*, 61(2) :180–191, 2009.

[20] T. Rafa. *Combinaison d'un Profil Social et un Profil Situationnel pour la Recherche d'Information*. Thèse de doctorat, Université, 2023.

[21] Alfred Kobsa. Generic user modeling systems. *User Modeling and User-Adapted Interaction*, 11(1-2) :49–63, 2001.

[22] Peter Ingwersen and Kalervo Järvelin. *The Turn : Integration of Information Seeking and Retrieval in Context*. Springer, 2005.

[23] Anind K. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, 5(1) :4–7, 2001.

[24] Peter Brusilovsky. Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 11(1-2) :87–110, 2001.

[25] Ourdia Ressad-Bouidghaghen. *Accès contextuel à l'information dans un environnement mobile : approche basée sur l'utilisation d'un profil situationnel de l'utilisateur et d'un profil de localisation des requêtes*. PhD thesis, Toulouse 3, 2011.

[26] Diane Kelly and Jaime Teevan. Implicit feedback for ir. *SIGIR Forum*, 37(2) :18–28, 2003.

[27] Mark Claypool et al. Implicit interest indicators. In *Proceedings of IUI*, pages 33–40, 2001.

[28] Ralf Kraft et al. Y !q : Contextual search at the point of inspiration. In *Proceedings of WWW Conference*, 2005.

[29] Alexander Sieg et al. Web search personalization with ontological user profiles. In *Proceedings of CIKM*, 2004.

[30] Fang Liu et al. Mining query-context pairs. In *Proceedings of WWW Conference*, 2004.

[31] Georgia Koutrika and Yannis Ioannidis. Personalized ir using relevance feedback network. In *Proceedings of CIKM*, 2005.

[32] Steve Fox et al. Evaluating implicit measures to improve web search. *ACM TOIS*, 23(2) :147–168, 2005.

[33] J. Ben Schafer et al. Collaborative filtering recommender systems. In *The Adaptive Web*, pages 291–324. Springer, 2007.

[34] Xiaoyuan Su and Taghi M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in AI*, 2009.

[35] Michael J. Pazzani and Daniel Billsus. Content-based recommendation systems. In *The Adaptive Web*, pages 325–341. Springer, 2007.

[36] Vijay Raghavan and Hakan Sever. On reusability of user profiles in personalized ir. In *Proceedings of SIGIR*, 1995.

[37] Henry Lieberman. Letizia : An agent that assists web browsing. In *Proceedings of IJCAI*, pages 924–929, 1995.

[38] Giorgio Gentili et al. Semantic user profiles for ir. In *Proceedings of KES Conference*, 2003.

[39] Michael J. Pazzani et al. Syskill and webert : Identifying interesting web sites. In *Proceedings of AAAI*, pages 54–61, 1996.

[40] Dimitar Kostadinov. *Multidimensional user modeling*. PhD thesis, Concordia University, 2008.

[41] Thanh H. Duong et al. Semantic personalization based on odp ontology. *Information Sciences*, 2013.

[42] Mariam Daoud et al. Semantic graph-based user profile for personalized search. *JASIST*, 2011.

[43] Daniela Godoy and Analía Amandi. User profiling in personal information agents. *Knowledge Engineering Review*, 20(4) :329–361, 2005.

[44] Ivan Koychev and Ingrid Schwab. Adaptation to drifting user interests. In *ECML Workshop*, 2000.

[45] Kazunari Sugiyama et al. Adaptive web search based on user profile. In *Proceedings of WWW Conference*, 2004.

[46] Bilel Moulahi. *Temporal and contextual dimensions in IR*. PhD thesis, Université Paul Sabatier, 2015.

[47] Pierre-Yves Vandenbussche et al. Temporal ir combining document and content time. In *Proceedings of ECIR*, 2011.

[48] Nattiya Kanhabua et al. Temporal semantic learning for event extraction. In *Proceedings of ECML/PKDD*, 2012.

[49] Imen Akermi et al. Personalized recommendation in mobile environment guided by a situational user profile. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015.

[50] N. Bila et al. Mobile user profile with location-based situations. In *Proceedings of UMAP*, 2008.

[51] Shoji Hattori et al. Personalized web search with location-based user profiles. In *Proceedings of CIKM*, 2007.

[52] F. Kamer et al. Social annotations for user profiling. In *Proceedings of ECIR*, 2014.

[53] Yassine Ould Amer et al. Conversation search in tweets combining textual and social relevance. In *Proceedings of ECIR*, 2015.

[54] Michael G. Noll and Christoph Meinel. Web search personalization via social bookmarking. In *Proceedings of ISWC*, 2007.

[55] Sheng Xu et al. Tag-based user profiling for personalized web search. In *Proceedings of WWW*, 2008.

[56] David Vallet et al. Personalized content retrieval using social tags. In *Proceedings of ECIR*, 2010.

[57] Paul Heymann et al. Social tag prediction. In *Proceedings of SIGIR*, 2008.

[58] Kerstin Bischoff et al. Tag motivation study. In *Proceedings of WWW Conference*, 2008.

[59] Nesrine Bouhini et al. Social annotations and profile terms for personalized search. In *Proceedings of ECIR*, 2016.

[60] Barry Smyth. Community-based web search. *AI Magazine*, 2011.

[61] Zohra Saoud and Samir Kechid. Distributed ir with social annotations and relations. In *Proceedings of CICLing*, 2016.

[62] Xavier Aimé et al. Semantic ir approaches. *Journal of Web Semantics*, 2010.

[63] K. M. Fouad et al. Semantic user profile for personalized ir. *IJCSI*, 2012.

[64] Hajer Zargayouna et al. Semantic resources for personalized ir. In *Proceedings of ECIR*, 2015.

[65] Ahmed Mohammed and Alia Abdelmoty. Semantic spatial profile for location-based social networks. In *GIR Workshop*, 2017.

[66] Ralf Schenkel et al. Social and semantic context for personalized search. In *Proceedings of ECIR*, 2008.

[67] Lynda Tamine. *Contextual IR : Models and evaluation*. Hdr, Université Paul Sabatier, 2010.

[68] Xuehua Shen et al. Implicit user modeling for search. In *Proceedings of SIGIR*, pages 307–314, 2005.

[69] Zhicheng Dou et al. A large-scale evaluation and analysis of personalized search. In *Proceedings of WWW Conference*, pages 581–590, 2007.

[70] W. Shafik et al. Social activity-based web personalization. In *Proceedings of ICTAI*, 2015.

[71] Lynda Lechani and Mohand Boughanem. Evaluation of personalized ir. In *Workshop on Adaptive IR*, 2005.

[72] Chris Buckley and Ellen M. Voorhees. Evaluating evaluation measure stability. In *Proceedings of SIGIR*, pages 33–40, 2000.

[73] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yinggong Min, Bowen Zhang, Jun Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv :2303.18223*, 2023.

[74] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33 :1877–1901, 2020.

[75] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm : Scaling language modeling with pathways. *arXiv preprint arXiv :2204.02311*, 2022.

[76] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2 : Open foundation and fine-tuned chat models. *arXiv preprint arXiv :2307.09288*, 2023.

[77] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Luc Saulnier, et al. Mistral 7b. *arXiv preprint arXiv :2310.06825*, 2023.

[78] Joseph Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1) :36–45, 1966.

[79] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[80] Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. Stanford University, 2023.

[81] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, pages 1715–1725, Berlin, Germany, 2016. Association for Computational Linguistics.

[82] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[83] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv :1607.06450*, 2016.

[84] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv :2303.08774*, 2023.

[85] Google Gemini Team. Gemini 1.5 : Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv :2403.05530*, 2024.

[86] Alec Radford, Karthik Narasimhan Wu, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. `https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf`, 2018. OpenAI.

[87] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140) :1–67, 2020.

[88] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization : Your language model is secretly a reward model. *Advances in neural information processing systems*, 36 :53728–53741, 2023.

[89] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora : Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[90] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, Luke Freivalds, and Luke Zettlemoyer. Qlora : Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems*, 2023.

[91] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 2022.

[92] Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models : A survey. In *Findings of the Association for Computational Linguistics : ACL 2023*, pages 13862–13896, 2023.

[93] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12) :1–38, 2023.

[94] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Tang, Xiaoxue Chen, Yukun Chen, Shuyan Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6) :186345, 2023.

[95] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Sen Jin, Enyu Zhou, et al. The rise and potential of large language model based agents : A survey. *arXiv preprint arXiv :2309.07864*, 2023.

[96] Lilian Weng. Llm-powered autonomous agents. `https://lilianweng.github.io/posts/2023-06-23-agent/`, 2023.

[97] Joon Sung Park, Joseph C O'Brien, Carrie J Cai, Meredith Ring Morris, Percy Liang, and Michael S Bernstein. Generative agents : Interactive simulacra of human behavior. *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–22, 2023.

[98] Timo Schick, Jane Dwivedi-Yu, Roberto Dürr, Sebastian Gehman, Scott R Bowman, Graham Durrett, and Luke Smith. Toolformer : Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 2023.

[99] Harrison Chase. Langchain. `https://github.com/langchain-ai/langchain`, 2022.

[100] Jerry Liu et al. Llamaindex : A data framework for llm applications. `https://github.com/run-llama/llama_index`, 2022.

[101] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Binhang Jiang, Xiaoyuan Zhang, Chunyuan Wang, Wenlong Zhang, et al. Autogen : Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv :2308.08155*, 2023.

[102] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React : Synergizing reasoning and acting in language models. *International Conference on Learning Representations*, 2023.

[103] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Tang, Xiaoxue Chen, Yukun Chen, Shuyan Lin, et al. Plan-and-solve prompting : Improving zero-shot chain-of-thought reasoning by large language models. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023.

[104] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion : Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 2023.

[105] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallahan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhu, Yiming Bhattacharyya, et al. Self-refine : Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 2023.

[106] Andres M Bran, Samuel M Cox, Andrew D White, and Philippe Schwaller. Chemcrow : Augmenting large-language models with chemistry tools. *Nature Machine Intelligence*, 5(6) :525–535, 2023.

[107] Sirui Hong, Xiawu Zheng, Jiaxin Chen, Yuheng Cheng, Ce Wang, Zili Zhang, Zihan Wang, Pengliang Zhang, Yu Cai, et al. Metagpt : Meta programming for a multi-agent collaborative framework. *International Conference on Learning Representations*, 2024.

[108] Xu Tang, Yunxiang Zhang, Jiacheng Zhou, Ziming Zhang, Haoran Wang, Xiaoxuan Chen, et al. Medagents : Large language models as collaborators for zero-shot medical reasoning. *arXiv preprint arXiv :2311.10537*, 2023.

[109] Xiao Liu, Songyang Yu, Zekun Zhang, Zhaoyu Yang, Zihao Miao, Xinyu Wang, Chang Wang, Jingneng Wang, Yu Wang, et al. Dynamic llm-agent network : An llm-agent collaboration framework with agent team optimization. *arXiv preprint arXiv :2310.02170*, 2023.

[110] Michael Ahn, Anthony Zhu, R. Martin Stone, Dragomir A. Lu, Constantin Peter, Dmitry Lu, et al. Do as i can, not as i say : Grounding language in robotic affordances. *Conference on Robot Learning*, 2022.

[111] Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *International Conference on Machine Learning*, 2024.

[112] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Qianyu Wang, and Haoran Sun. Retrieval-augmented generation for large language models : A survey. *arXiv preprint arXiv :2312.10997*, 2024.

[113] Yutao Zhu, Haonan Liu, Chen Liu, Weinan Gao, Jian Sun, and Jiawei Yang. Large language models for information retrieval : A survey. *arXiv preprint arXiv :2308.07107*, 2023.

[114] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, 2020.

[115] Liang Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Tang, Xiaoxue Chen, Yukun Chen, Shuyan Lin, et al. Text embeddings by weakly-supervised contrastive pre-training. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

[116] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023.

[117] Amir Salemi and Hamed Zamani. Towards personalized retrieval-augmented generation. *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024.

[118] Krisztian Balog and Tom Kenter. Personal knowledge graphs : A research agenda. *Proceedings of the 2019 Conference on Creativity and Cognition*, 2019.

[119] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 2020.

[120] Vaibhav Adlakha, Xingdi Wang, Yiming Li, Chenyan Zhang, Wen-tau Yih, and Mike Lewis. Evaluating correctness and faithfulness of instruction-following models for question answering. *Transactions of the Association for Computational Linguistics*, 2024.

[121] Ke Mao et al. Rag-driver : Generalisable driving explanations with retrieval-augmented in-context learning of foundation models, feb 2024.

[122] Xueyao Ma, Xinyu Wang, Siwei Zhang, Xiaojun Wang, Hao Wang, et al. Query rewriting for retrieval-augmented large language models. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[123] Luyu Gao, Jamie Callan, Jamie Lester, Lingyao Yang, Ji Wei, et al. Precise zero-shot dense retrieval without relevance labels. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023.

[124] Ofir Press, Murong Zhang, Sewon Min, Gabriel Schmidt, Noah A Smith, and Hal Daumé III. Measuring and narrowing the compositionality gap in language models. *Findings of the Association for Computational Linguistics : EMNLP 2023*, 2023.

[125] Jimmy Lin. A few brief notes on deepimpact, coil, and a conceptual framework for information retrieval techniques. *arXiv preprint arXiv :2106.14807*, 2022.

[126] Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 258–265, 2009.

[127] Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with bert. *arXiv preprint arXiv :1901.04085*, 2019.

[128] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yinuo Zhuang, Zihao Lin, Zhuohan Saleh, Chaowei Li, Dacheng Li, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 2023.

[129] Harsh Trivedi et al. Ircot : Interleaving retrieval with chain-of-thought reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, pages 5504–5519, Toronto, Canada, 2023. Association for Computational Linguistics.

[130] Zhiheng Shao, Yeyun Gong, Yelong Zhang, Xinting Li, Jing Jiang, Chen Song, Jie Zhao, Shuai Yang, Nan Duan, and Ming Zhou. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. *Findings of the Association for Computational Linguistics : EMNLP 2023*, 2023.

[131] Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. Active retrieval augmented generation. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[132] Zihan Wu et al. Personalized retrieval-augmented generation for dialogue systems. *arXiv preprint arXiv :2406.12580*, 2024.

[133] Tefko Saracevic. Relevance : A review of and a framework for the thinking on the notion in information science. *Journal of the American Society for Information Science*, 26(6) :321–343, 1975.

[134] Zhihan Zhang et al. Towards long-term memory in llm agents : A survey. *arXiv preprint arXiv :2404.01562*, 2024.

[135] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt : Browser-assisted question-answering with human feedback. *arXiv preprint arXiv :2112.09332*, 2022.

[136] Jia Chen et al. Benchmarking large language models in retrieval-augmented generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

[137] Sachin Es et al. Ragas : Automated evaluation of retrieval-augmented generation. *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics : System Demonstrations*, 2024.

[138] Yunfan Gao et al. Modular rag : Transforming rag systems into lego-like reconfigurable frameworks. *arXiv preprint arXiv :2407.21059*, 2023.